Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Double-authentication-preventing signatures revisited: new definition and construction from chameleon hash[*]

Fei LI[†1,2,3], Wei GAO[1,2,3], Gui-lin WANG[4], Ke-fei CHEN[5], Chun-ming TANG[2]

[1]*School of Mathematics and Statistics, Ludong University, Yantai 264025, China*
[2]*Key Laboratory of Information Security, Guangzhou University, Guangzhou 510006, China*
[3]*China-USA Computer Science Research Center, Nanjing University of Information Science and Technology, Nanjing 210044, China*
[4]*Shield Lab, Singapore Research Center of Huawei, 117674, Singapore*
[5]*Department of Mathematics, Hangzhou Normal University, Hangzhou 311121, China*
[†]E-mail: miss_lifei@163.com

Received Jan. 3, 2017; Revision accepted May 22, 2017; Crosschecked Jan. 22, 2019

Abstract: Double-authentication-preventing signature (DAPS) is a novel signature notion proposed at ESORICS 2014. The double-authentication-preventing property means that any pair of signatures on two different messages with the same subject will result in an immediate collapse of the signature system. A few potential applications of DAPS have been discussed by its inventors, such as providing a kind of self-enforcement to discourage certificate authority (CA) from misbehaving in public key infrastructure and offering CA some cryptographic arguments to resist legal coercion. In this study, we focus on some fundamental issues on DAPS. We propose a new definition, which is slightly weakened but still reasonable and strong enough to capture the DAPS concept. We develop the new notion of invertible chameleon hash functions with key exposure. Then we propose a generic DAPS scheme, which is provably secure if the underlying invertible chameleon hash function with key exposure is secure. We instantiate this general construction to obtain the DAPS schemes respectively based on the well-known assumptions of integer factorization, Rivest-Shamir-Adleman (RSA), and computational Diffie-Hellman (CDH). They are more efficient than previous DAPS schemes. Furthermore, unlike previous constructions, the trusted setup condition is not needed by our DAPS schemes based on RSA and CDH.

Key words: Double-authentication-preventing signatures; Chameleon hash function; Digital signature; Provable security; Authority trust level
https://doi.org/10.1631/FITEE.1700005 **CLC number:** TP309

1 Introduction

The novel notion of double-authentication-preventing signature (DAPS) was first proposed at ESORICS 2014 (Poettering and Stebila, 2014, 2017). For DAPS, the signed value consists of the subject part and the message part. It aims to provide a certain deterrent mechanism for preventing the signer from signing more than one message for each subject. In particular, if two different messages with the same subject are signed, the signer will be immediately and automatically punished. For example, the punishment may be the automatic exposure of the signing key. Since this punishment can be seen as fatal for the signer, it forms one deterrent mechanism for ensuring this so-called double-

[*] Project supported by the National Natural Science Foundation of China (Nos. 61202475, 61133014, and 61472114), the Science and Technology Planning Key Project of Shandong Universities, China (No. J18KA326), and the Science and Technology Planning Key Project of Guangdong Province, China (No. 2016B010124014)
ORCID: Fei LI, http://orcid.org/0000-0002-8916-8330

authentication-preventing requirement.

The practical applicability of DAPS has been carefully discussed by Poettering and Stebila (2014). In particular, the authors proposed and examined some application cases in detail. Here, we review the potential applications of DAPS for certificate authorities in public key infrastructure (PKI). First, DAPS can help the PKI system realize an authority trust level higher than three (Girault, 1991). When using a traditional signature to generate certificates, authority trust is at level three: although certificate authority (CA) can generate a certificate binding any malicious public key, there is no way for CA to deny the proof of this malicious behavior.

The trust level above has been thought of as the highest possible in cryptography since being defined by Girault (1991). However, it ensures only the proof, but not any final penalty, the execution of which requires a third-party arbitrator. In contrast, using DAPS to generate certificates, a higher authority trust level may be obtained, as described below: although CA can generate a certificate binding any malicious public key, there is no way for CA to avoid an objective, automatic, and immediate penalty for this misbehavior.

Note that the above idea has been implicitly proposed in Poettering and Stebila (2014) and we tentatively call it authority trust level four. DAPS can help the honest signer (CA in PKI) resist double-signature coercion in the form of legal demands from governments, such as the so-called compelled certificate creation attack (Soghoian and Stamm, 2011; Poettering and Stebila, 2014), with the argument that this co-operation will fatally expose CA's signing key. In fact, as demonstrated by the Edward Snowden incident, governments may be very interested in such compelling behaviors (Soghoian and Stamm, 2011). Additionally, as discussed by the inventors (Poettering and Stebila, 2014), in practice as a new cryptographic primitive, the existence of some imperfections in these DAPS applications is unavoidable. In other words, what DAPS contributes to these applications is not the full practical solution, but a theoretical starting point for a better practical solution than currently exists. For details on potential applications, please refer to Poettering and Stebila (2014).

Poettering and Stebila (2014) presented the formalization for DAPS, constructed one generic DAPS scheme from the new cryptographic primitive called extractable $2:1$ trapdoor functions and the unique instantiation based on sign-agnostic quadratic residues. They also discussed some applications of DAPS in PKI and time-stamping. In Poettering and Stebila (2014), many cryptographic primitives which seem relative to DAPS are compared with DAPS, including traitor tracing (Chor et al., 2000), accountable identity based encryption (Goyal, 2007), digital cash schemes (Chaum et al., 1988), one-time signatures (Mohassel, 2010), fail-stop signatures (Pedersen and Pfitzmann, 1997), and chameleon hash functions (Krawczyk and Rabin, 2000). In particular, the authors put forth the open problem of how DAPS could be constructed from chameleon hash functions.

The research on DAPS has just begun and there are many interesting problems on this topic. In their pioneering work (Poettering and Stebila, 2014), the authors stated some basic problems on DAPS in theory and in practice. In fact, one generic DAPS construction from the so-called extractable $2:1$ trapdoor functions was proposed in Poettering and Stebila (2014). However, there is only one instantiation scheme based on the integer factorization assumption. Furthermore, this concrete DAPS signature scheme has a much larger signature size and longer running time for signing operations, compared with common signature schemes such as Rivest-Shamir-Adleman (RSA) signatures. Additionally, although the authors have formally defined the double-signature forgeability with or without a trusted setup, there is still no DAPS scheme that is secure in the untrusted setup model, without dependence on the zero-knowledge techniques.

In this study, we focus on some fundamental issues of DAPS in both theory and practice, including, but not limited to, the above problems. Our contributions are as follows:

1. For DAPS, we revisit the most basic definition of double-signature forgeability, which formally captures the double-authentication-preventing (DAP) property. We find that the original definition uses conditions that are too severe to capture the DAP property. Accordingly, we propose a new definition for double-signature forgeability. It is slightly weaker than the previous one, but still reasonable since it remains strong enough to ensure the DAP property. As will be seen, under this new definition,

it becomes easier to realize DAPS schemes.

2. For chameleon hash functions, we study the properties of invertibility and key exposure. As will be seen, although the invertibility is rarely mentioned for chameleon hash functions in cryptography, it plays a key role for our generic construction. We propose a new notion of the invertible chameleon hash function with key exposure. We show that such new kinds of chameleon hash functions can be easily constructed based on integer factorization (IF), RSA, and computational Diffie-Hellman (CDH).

3. Depending upon the above new formalizations of DAPS and chameleon hash functions, we propose a provably secure generic framework for constructing the DAPS scheme from any invertible chameleon hash function with key exposure. This framework for constructing DAPS is much more generic than the previous one in Poettering and Stebila (2014), since the chameleon hash function is much more popular than the new cryptographic primitive, extractable 2 : 1 trapdoor functions.

4. Following the generic DAPS scheme, we instantiate some more efficient DAPS schemes based on IF, RSA, and CDH, respectively. Compared with the first DAPS scheme (Poettering and Stebila, 2014), the DAP property in the untrusted setup model (Poettering and Stebila, 2014) for our RSA/CDH-based construction is realized without depending on zero-knowledge techniques.

2 Double-authentication-preventing signature (DAPS)

In this section, we present an improved definition for DAPS. In particular, the difference from that stated in Poettering and Stebila (2014) is the new formalization for the double-signature forgeability to capture the DAP property (Definition 4). This is the basis of our new generic DAPS construction.

Definition 1 (DAPS (Poettering and Stebila, 2014)) A double-authentication-preventing signature scheme has three basic algorithms, DAPS = (SKg, SSign, SVrf):

1. $\mathrm{SKg}(1^k) \to (\mathrm{sk}, \mathrm{pk})$. Given security parameter 1^k, it outputs secret/public keys sk/pk.

2. $\mathrm{SSign}(\mathrm{sk}, \mathrm{sbj}, \mathrm{msg}) \to \sigma$. Given secret key sk, subject sbj, and message msg, it outputs signature σ.

3. $\mathrm{SVrf}(\mathrm{pk}, \mathrm{sbj}, \mathrm{msg}, \sigma) \to b \in \{0, 1\}$. Given public key pk, subject sbj, message msg, and signature σ, it outputs $b = 1$ if σ is valid, and $b = 0$ if σ is invalid.

Definition 2 (Correctness (Poettering and Stebila, 2014)) A DAPS scheme is correct if, for any key pair $(\mathrm{sk}, \mathrm{pk}) \leftarrow \mathrm{SKg}(1^k)$, any subject sbj, any message msg, and any signature $\sigma \leftarrow \mathrm{SSign}(\mathrm{sk}, \mathrm{sbj}, \mathrm{msg})$, it always holds that $\mathrm{SVrf}(\mathrm{pk}, \mathrm{sbj}, \mathrm{msg}, \sigma) = 1$.

Definition 3 (Existential unforgeability (Poettering and Stebila, 2014)) A DAPS scheme is existentially unforgeable, if for any probabilistic polynomial-time adversary F with oracle access to signing oracle $O^{\mathrm{Sign}}(\cdot)$, the following game returns 1 with negligible probability:

1. $(\mathrm{sk}, \mathrm{pk}) \leftarrow \mathrm{SKg}(1^k)$.

2. $(\mathrm{sbj}^*, \mathrm{msg}^*, \sigma^*) \leftarrow F^{O^{\mathrm{Sign}}(\cdot)}(\mathrm{pk})$.

3. Return 1 if all the following conditions hold: (1) $\mathrm{SVrf}(\mathrm{pk}, \mathrm{sbj}^*, \mathrm{msg}^*, \sigma^*) = 1$; (2) $(\mathrm{sbj}^*, \mathrm{msg}^*) \notin L$, where L denotes the set of all pairs (sbj, msg) which have been signed by oracle $O^{\mathrm{Sign}}(\cdot)$; (3) for any two pairs in L denoted by $(\mathrm{sbj}_1, \mathrm{msg}_0)$ and $(\mathrm{sbj}_1, \mathrm{msg}_1)$, it holds that $\mathrm{sbj}_0 \neq \mathrm{sbj}_1$.

Definition 4 (Double-signature forgeability, double-authentication-preventing property) A DAPS scheme is double-signature forgeable in the untrusted setup model, if there exists an algorithm SForge such that for any adversary A, the following experiment returns 0 with negligible probability:

1. Untrusted setup: $(\mathrm{sk}, \mathrm{pk}, (\mathrm{sbj}, \mathrm{msg}, \sigma), (\mathrm{sbj}, \widehat{\mathrm{msg}}, \hat{\sigma}), (\mathrm{sbj}^*, \mathrm{msg}^*, \sigma^*), \widehat{\mathrm{msg}}^*) \leftarrow A(1^\lambda)$. Trusted setup: $(\mathrm{sk}, \mathrm{pk}) \leftarrow \mathrm{SKg}(1^k)$, $((\mathrm{sbj}, \mathrm{msg}, \sigma), (\mathrm{sbj}, \widehat{\mathrm{msg}}, \hat{\sigma}), (\mathrm{sbj}^*, \mathrm{msg}^*, \sigma^*), \widehat{\mathrm{msg}}^*) \leftarrow A(\mathrm{sk}, \mathrm{pk})$. Here, it is required that σ, $\hat{\sigma}$, σ^* should be valid signatures.

2. $(\mathrm{sbj}^*, \widehat{\mathrm{msg}}^*, \hat{\sigma}^*) \leftarrow \mathrm{SForge}(\mathrm{pk}, (\mathrm{sbj}, \mathrm{msg}, \sigma), (\mathrm{sbj}, \widehat{\mathrm{msg}}, \hat{\sigma}), (\mathrm{sbj}^*, \mathrm{msg}^*, \sigma^*), \widehat{\mathrm{msg}}^*)$.

3. Return 0 if $\mathrm{SVrf}(\mathrm{pk}, \mathrm{sbj}^*, \widehat{\mathrm{msg}}^*, \hat{\sigma}^*) = 0$.

By contrast, in Poettering and Stebila (2017), the DAP property was formalized by the algorithm Forge for the double-signature forgeability or Extract for the double-signature extractability as follows:

1. $\mathrm{Forge}(\mathrm{pk}, (\mathrm{sbj}, \mathrm{msg}, \sigma), (\mathrm{sbj}, \widehat{\mathrm{msg}}, \hat{\sigma}), \mathrm{sbj}^*, \mathrm{msg}^*) \to (\mathrm{sbj}^*, \mathrm{msg}^*, \sigma^*)$. The inputs are two valid subject/message/signature triples (sbj, msg, σ), (sbj, $\widehat{\mathrm{msg}}$, $\hat{\sigma}$), and any subject/message (sbj*, msg*), and the output is a new valid triple (sbj*, msg*, σ^*).

2. $\mathrm{Extract}(\mathrm{pk}, (\mathrm{sbj}, \mathrm{msg}, \sigma), (\mathrm{sbj}, \widehat{\mathrm{msg}}, \hat{\sigma})) \to \mathrm{sk}$. The inputs are two valid signature/subject/message triples (sbj, msg, σ), (sbj,

$\widehat{\text{msg}}$, $\hat{\sigma}$) and the output is secret key sk.

Now we show some observations on these three algorithms (Extract, Forge, and SForge) used for capturing the DAP property:

1. Extract is functionally equivalent to Forge, since they both mean that any forgery can be generated for any message and any subject, once two different signatures on the same subject are known.

2. With respect to input parameters, SForge is weaker than Forge, since SForge additionally needs one signature for the attacked subject as inputs.

3. With respect to output parameters, SForge remains strong enough to deter the double-signature misbehavior or ensure the DAP property, although it is weaker than Forge. For example, assume that CA generates a legal certificate (DAPS) binding a user identity (subject) with a public key (message) and then generates a malicious certificate binding this identity with a malicious public key. According to SForge, this malicious behavior will result in that the certificate binding "any" identity with "any" public key can be forged only if one certificate for this identity has been issued. In other words, all certificates, previous or future ones, will never worth trusting. Of course, this is enough to make the CA collapse.

As will be seen, this relaxed definition makes it easier to construct DAPS and can be seen as the base for our generic DAPS construction.

3 Invertible chameleon hash functions with key exposure

In this section, we present the new notion of invertible chameleon hash functions with key exposure. In particular, motivated by the open problem of whether DAPS can be constructed from the chameleon hash function (Poettering and Stebila, 2014), we discover that two significant properties are very important for solving this problem. One is the rarely mentioned property of invertibility. The other is the well studied key exposure property.

The chameleon hash function (Krawczyk and Rabin, 2000) is collision-resistant for the person holding only the public key, and chameleon for the owner of the trapdoor information who can change the input to the function into any value of his/her choice without changing the output. The key exposure problem for chameleon hash functions (Ateniese and de Medeiros, 2004b) means that two different inputs with the same hash value can be used to compute the trapdoor information. For practical applications, it is usually desirable to design the chameleon hash function without key exposure.

Definition 5 (Invertible chameleon hash function with key exposure) An invertible chameleon hash function with key exposure is the tuple of five algorithms ICHF = {ChKg, ChHash, ChCld, ChIvt, ChExp}, where the chameleon property, key exposure property, and invertible property are formalized by the existence of ChCld, ChExp, and ChIvt, respectively:

1. Key generation. ChKg(1^k) outputs hash key chk, collision trapdoor ctd, and inversion trapdoor itd.

2. Hash evaluation. ChHash(chk, m, r) outputs hash value h of message m with randomness r.

3. Chameleon property. ChCld(ctd, m, r, $\hat{m}$) outputs a randomness $\hat{r}$ such that ChHash(chk, m, r) = ChHash(chk, $\hat{m}$, $\hat{r}$), where $m \neq \hat{m}$.

4. Key exposure property. ChExp(chk, m, r, $\hat{m}$, $\hat{r}$) outputs collision trapdoor ctd, where $m \neq \hat{m}$, ChHash(chk, m, r) = ChHash(chk, $\hat{m}$, $\hat{r}$).

5. Invertible property. ChIvt(itd, h, m) outputs randomness r such that ChHash(chk, m, r) = h.

Definition 6 (Collision resistance (Krawczyk and Rabin, 2000)) An invertible chameleon hash function with key exposure ICHF is collision resistant, if for any probabilistic polynomial-time adversary A, it always holds that for negligible ϵ, shown as

$$\Pr\begin{pmatrix} (\text{chk}, \text{ctd}, \text{itd}) \leftarrow \text{ChKg}(1^k); \\ ((m, r), (\hat{m}, \hat{r})) \leftarrow A(\text{chk}) : \\ \text{ChHash}(\text{chk}, m, r) = \text{ChHash} \\ (\text{chk}, \hat{m}, \hat{r}), \wedge (m, r) \neq (\hat{m}, \hat{r}) \end{pmatrix} < \epsilon. \quad (1)$$

Now we further discuss the above new notion, by taking it as the special chameleon hash function with the two additional properties of invertibility and key exposure.

First, we discuss the key exposure property. Conceptually speaking, the key exposure property is not a new notion and has been seen as undesirable for chameleon hash functions, as first mentioned in Ateniese and de Medeiros (2004a, 2004b). Many efforts were made to design chameleon hash functions without key exposure, just as done in Ateniese and de Medeiros (2004a, 2004b), Chen et al. (2004, 2007, 2011, 2014), and Gao et al. (2007, 2009). On the other hand, many chameleon hash functions almost have the key exposure property. In the above definition, this traditionally undesirable property is "positively" put forth and will play an important role in designing DAPS schemes. For the above definition, the collision pair $((m, r), (\hat{m}, \hat{r}))$ does not expose the inversion (main) trapdoor, but exposes the collision (secondary) trapdoor. In other words, with respect to the main trapdoor, they are exposure-free; with respect to the secondary trapdoor, they have the key exposure property.

Next, we turn to the invertibility. As far as we know, it is the first time invertibility has been explicitly considered for chameleon hash functions. In fact, it can be seen that the inverting function ChIvt is the inverse function for ChHash. Formally speaking, according to Bellare et al. (1998), function family $\{\text{ChHash}(\text{chk}, \cdot) : (m, r) \mapsto \text{ChHash}(\text{chk}, \cdot)\}_{\text{chk}}$ is just the many-to-one trapdoor one-way function. In particular, if m is further fixed, function family $\{\text{ChHash}(\text{chk}, m, \cdot) : r \mapsto \text{ChHash}(\text{chk}, m, \cdot)\}_{\text{chk},m}$ is just the trapdoor one-way permutation. Following this observation, the invertible property can be seen as one common notion in cryptography, despite rarely being mentioned for chameleon hash functions.

Third, we discuss the advantage of invertible chameleon hash functions without key exposure. Note that a cryptographic function is usually thought to be more desirable. The more difficult the task that the trapdoor holder can solve is, the task that the entity without the trapdoor cannot solve is easier. It thus follows that an invertible chameleon hash function without key exposure is stronger than a trapdoor one-way function, since the task that the entity without the trapdoor cannot solve for the former (to find a collision) is easier than that for the latter (to invert one function image). On the other hand, an invertible chameleon hash function without key exposure is stronger than a classical chameleon hash function, since the task that the trapdoor holder can solve for the former (to invert one function image) is more difficult than that for the latter (to find a collision). Hence, as a fundamental cryptographic primitive, an invertible chameleon hash function without key exposure has better cryptographic properties than the classical notion of trapdoor one-way functions and chameleon hash functions. Furthermore, these better cryptographic properties make it easier to design cryptographic schemes. The current task to design a DAPS scheme based on an invertible chameleon hash function without key exposure is just one example.

At last, we claim that many (but not all) existing chameleon hash functions are invertible and with key exposure. As will be shown in Section 6, three existing chameleon hash functions based on IF, RSA, and CDH "naturally" have the invertibility and key exposure properties.

4 Generic double-authentication-preventing signature construction from chameleon hash functions

Motivated by the open problem of whether DAPS could be constructed from chameleon hash functions (Poettering and Stebila, 2014), we try to explore the theoretical relation between DAPS and chameleon hash functions. On the one hand, to make the construction task theoretically easier, we try to relax the formal condition for capturing the DAP property as demonstrated in Section 2. On the other hand, to make the basic tool more powerful, we explore additional properties for the chameleon hash functions as outlined in Section 3. As a result, we succeed in proposing the following general DAPS construction, and hence solve the open problem (Poettering and Stebila, 2014):

Given the invertible chameleon hash scheme with key exposure ICHF = {ChKg, ChHash, ChCld, ChExp, ChIvt}, the DAPS scheme and the associated algorithm SForge are constructed as follows (additionally, it needs two cryptographic hash functions, $H_1(\cdot)$ and $H_2(\cdot)$, which respectively hash the input value into the hash value space and the message space of ICHF):

1. SKg(1^k). Run (chk, ctd, itd) $\leftarrow$ ChKg(1^k), and return public key pk = chk and secret key sk = (ctd, itd). Additionally, we assume that the random seed length l is commonly known and not explicitly included in the public key.

2. SSign(pk, sk, sbj, msg). For sk = (ctd, itd) and pk = chk, randomly select $s \leftarrow_R \{0,1\}^l$, compute $h \leftarrow H_1(\text{sbj})$, $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$, $r \leftarrow$ ChIvt(itd, h, m), and return signature $\sigma \leftarrow (r, s)$.

3. SVrf(pk, sbj, msg, σ). For pk = chk, $\sigma = (r, s)$, compute $h \leftarrow H_1(\text{sbj})$, $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$, and return 1 if $h =$ ChHash(chk, m, r) or 0 otherwise.

4. SForge(pk, (sbj, msg, σ), (sbj, $\widehat{\text{msg}}$, $\hat{\sigma}$), (sbj*, msg*, σ^*), $\widehat{\text{msg}}^*$). (1) For pk = chk, $\sigma = (r, s)$, $\hat{\sigma} = (\hat{r}, \hat{s})$, and $\sigma^* = (r^*, s^*)$, select $\hat{s}^* \leftarrow_R \{0,1\}^l$ randomly, and set $h \leftarrow H_1(\text{sbj})$, $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$, $\hat{m} \leftarrow H_2(\text{sbj}||\widehat{\text{msg}}||\hat{s})$, $m^* \leftarrow H_2(\text{sbj}^*||\text{msg}^*||s^*)$, $\hat{m}^* \leftarrow H_2(\text{sbj}^*||\widehat{\text{sbj}}^*||\hat{s}^*)$. (2) Compute ctd $\leftarrow$ ChExp(chk, $h, m, r, \hat{m}, \hat{r}$), $\hat{r}^* \leftarrow$ ChCld(ctd, m^*, r^*, $\hat{m}^*$). (3) Return (sbj*, $\widehat{\text{msg}}^*$, $\hat{\sigma}^*$), where $\hat{\sigma}^* = (\hat{r}^*, \hat{s}^*)$.

For this generic result, we present some observations as follows:

1. The generic DAPS scheme has provable security in the random oracle model as in Section 5.

2. The generic DAPS scheme has many instantiations better than the previous one in terms of computation efficiency, signature size, and cryptographic assumptions as in Section 6.

3. The instantiations based on RSA and CDH succeed in ensuring the untrusted setup model without needing the complicated zero-knowledge proof as in Section 7.

4. It is theoretically significant that the new cryptographic primitive of DAPS is almost immediately related with the popular primitive of chameleon hash functions.

5. As the new cryptographic primitive specially developed here, the invertible chameleon hash function with key exposure may have more applications in cryptographic theory and practice.

5 Security proof for generic double-authentication-preventing signature construction

In this section, we prove that the above generic DAPS scheme is double-signature-preventing (double-signature forgeable, as formalized in Definition 4) and existentially unforgeable (Definition 3) in the random oracle model, only if the underlying invertible chameleon hash function with key exposure is secure as formalized in Section 3.

Theorem 1 Following the key exposure property and the chameleon property of the underlying chameleon hash scheme ICHF = {ChKg, ChHash, ChCld, ChExp, ChIvt}, the DAPS scheme (SKg, SSign, SVrf) from ICHF is double-signature forgeable, with H_1, H_2 taken as random oracles.

Proof For the experiment in Definition 4, tuples (sbj, msg, σ), (sbj, $\widehat{\text{msg}}$, $\hat{\sigma}$), (sbj*, msg*, σ^*) provided by adversary A for SForge satisfy

$$\begin{cases} \text{SVrf(pk, } \sigma, \text{ sbj, msg)} = 1, \\ \text{SVrf(pk, } \hat{\sigma}, \text{ sbj, } \widehat{\text{msg}}) = 1, \\ \text{SVrf(pk, } \sigma^*, \text{ sbj}^*, \text{ msg}^*) = 1. \end{cases} \tag{2}$$

Then by the description of SVrf in DAPS, we have

$$\begin{cases} h = \text{ChHash}(\text{chk}, m, r), \\ h = \text{ChHash}(\text{chk}, \hat{m}, \hat{r}), \\ h^* = \text{ChHash}(\text{chk}, m^*, r^*), \end{cases} \tag{3}$$

where

$$\begin{cases} \text{chk} = \text{pk}, \ \sigma = (r, s), \\ \hat{\sigma} = (\hat{r}, \hat{s}), \ \sigma^* = (r^*, s^*), \\ h = H_1(\text{sbj}), \ h^* = H_1(\text{sbj}^*), \\ m = H_2(\text{sbj}||\text{msg}||s), \\ \hat{m} = H_2(\text{sbj}||\widehat{\text{msg}}||\hat{s}), \\ m^* = H_2(\text{sbj}^*||\text{msg}^*||s^*). \end{cases} \tag{4}$$

Then by the key exposure property of ICHF, if $m \neq \hat{m}$, then collision trapdoor key ctd can be computed: ctd $\leftarrow$ ChExp(chk, m, r, $\hat{m}$, $\hat{r}$).

Remark 1 Here, we discuss the untrusted/trusted setup model with respect to ChExp. If ChExp can extract the collision trapdoor with respect to the arbitrarily generated hash key, the key exposure property holds in the untrusted setup model. If ChExp can extract the collision trapdoor with respect to only the honestly generated hash key, the key exposure property holds in the trusted setup model. Furthermore, it can be seen that when the exposure property for ICHF holds in the trusted (or untrusted) model, the forgeability for DAPS accordingly holds in the trusted (or untrusted) model.

Then by the chameleon property of ICHF, for $\hat{m}^* = H_2(\text{sbj}^*||\widehat{\text{msg}}^*||\hat{s}^*)$, where $\hat{s}^* \leftarrow_R \{0,1\}^l$, corresponding randomness $\hat{r}^*$ can be computed, $\hat{r}^* \leftarrow$ ChCld(ctd, m^*, r^*, $\hat{m}^*$), such that $h^* =$ ChHash(chk, $\hat{m}^*$, $\hat{r}^*$). By the description of SVrf in DAPS, signature $\hat{\sigma}^* = (\hat{r}^*, \hat{s}^*)$ on subject/message (sbj*, $\widehat{\text{msg}}^*$) is valid, since $h^* =$ ChHash(chk, $\hat{m}^*$, $\hat{r}^*$).

All in all, if $m \neq \hat{m}$, SForge can generate valid tuple (sbj*, $\widehat{\text{msg}}^*$, $\hat{\sigma}^*$). In the random oracle model, we have $\Pr[H_2(\text{sbj}||\text{msg}||s) = H_2(\text{sbj}||\widehat{\text{msg}}||\hat{s})] \leq q(q-1)/2^{l+1}$, where l is the length of random seeds and q is the number of queries to random oracle

H_2 during this game. Hence, SForge can generate valid tuple (sbj.000-0.*, $\widehat{\text{msg}}^*$, $\hat{\sigma}^*$) with a probability greater than $1 - q(q-1)/2^{l+1}$. Note that $q(q-1)/2^{l+1}$ is negligible for an l large enough. The proof is completed.

Theorem 2 Following the collision resistance property of the underlying chameleon hash scheme ICHF = {ChKg, ChHash, ChCld, ChExp, ChIvt}, the DAPS scheme (SKg, SSign, SVrf) from ICHF is existentially unforgeable, with H_1, H_2 taken as random oracles.

Proof Assume that there exists a forgery adversary F, as defined in Definition 3, against the DAPS scheme with success probability ϵ and running time t. This proof constructs colliding adversary A, as in Definition 6, against the chameleon hash scheme ICHF as follows:

First, A obtains hash key chk from its challenger, sets signature public key pk = chk, and sends pk to F. A sets $f = 0$ to record the current number of non-reputed queries on oracle O^{H_1}.

A simulates two random oracles O^{H_1}, O^{H_2} as follows. To answer query O^{H_1}(sbj), if this query has been previously answered, A just returns the previous answer. Otherwise, A randomly chooses m, r, sets $H_1(\text{sbj}) \leftarrow \text{ChHash(chk}, m, r)$, $f \leftarrow f+1$, and returns H_1(sbj). To answer random oracle query $O^{H_2}(\text{sbj}||\text{msg}||s)$, if this query has been previously answered, A just returns the previous answer. Otherwise, A randomly chooses one element c from the range of H_2 and returns $H_2(\text{sbj}||\text{msg}||s) = c$.

A simulates the signing oracle as follows: without loss of generality, assume that the subjects whose signatures are queried are all different. To simulate signing query O^{Sign}(sbj, msg), A obtains H_1(sbj) by making random oracle query O^{H_1}(sbj). According to the simulation of O^{H_1}, there must exist one tuple (m, r) such that $H_1(\text{sbj}) = \text{ChHash(chk}, m, r)$. A randomly chooses $s \leftarrow \{0,1\}^l$. For this s, if $H_2(\text{sbj}||\text{msg}||s)$ has been queried, A just terminates the game and outputs failure. Otherwise, it sets $H_2(\text{sbj}||\text{msg}||s) = m$ and uses this value as a simulation of O^{H_2}. A sets signature $\sigma \leftarrow (r, s)$ and sends it to F.

At last, F returns signature σ^* on subject sbj* and message msg*. If SVrf(pk, σ^*, sbj*, msg*) $\neq 1$, A terminates the game and declares failure. If SVrf(pk, σ^*, sbj*, msg*) $= 1$, by the description of SVrf in DAPS, it holds that $h^* = \text{ChHash(chk}, m^*, r^*)$, for $\sigma^* = (r^*, s^*)$, $m^* = H_2(\text{sbj}^*||\text{msg}^*||s^*)$, $h^* = H_1(\text{sbj}^*)$. By the simulation of $H_1(\text{sbj}^*)$, we have another pair ($\hat{m}^*$, $\hat{r}^*$) such that ChHash(chk, m^*, r^*) = ChHash(chk, $\hat{m}^*$, $\hat{r}^*$). Now A succeeds in resisting the collision resistance against ICHF.

According to the construction of our generic DAPS scheme, the above simulation is completely the same to the true game, only if A does not terminate the game when $H_2(\text{sbj}||\text{msg}||s)$ has been queried before signature query O^{Sign}(sbj, msg). The random choice of s ensures that this case happens with a probability less than

$$\sum_{i=0}^{q_\text{s}-1} \frac{q_2 + i}{2^l} = \frac{q_\text{s}(2q_2 + q_\text{s} - 1)}{2^{l+1}}, \tag{5}$$

where q_2 and q_s are the numbers of queries to random oracle O^{H_2} and signing oracle O^{Sign} respectively during this game. Additionally, we have just shown that if F returns the valid signature forgery, then A can obtain (m^*, r^*), ($\hat{m}^*$, $\hat{r}^*$) with the same hash value. Hence, in this simulation, A succeeds in presenting the collision with the probability Pr(A succeeds) > Pr(F succeeds)$-q_\text{s}(2q_2 + q_\text{s} - 1)/2^{l+1}$. The proof is completed.

6 Double-authentication-preventing signature instantiation based on integer factorization (IF), Rivest-Shamir-Adleman (RSA), and computational Diffie-Hellman (CDH)

6.1 DAPS based on IF

Recall that the pair of permutations ($f_0(\cdot)$, $f_1(\cdot)$) over the same domain are said to be claw-free trapdoor permutations (Goldwasser et al., 1988). If given only $f_0(\cdot)$, $f_1(\cdot)$, it is computationally infeasible to find a "claw" which is one pair (x_0, x_1) such that $f_0(x_0) = f_1(x_1)$ and there exist two efficient algorithms (f_0^{-1}, f_1^{-1}) inverting $f_0(\cdot)$, $f_1(\cdot)$ respectively. For formal details on this notion, refer to Goldwasser et al. (1988). Let $n = pq$, where p, q are two primes such that $p = 3 \mod 8$, $q = 7 \mod 8$. We review the following results in Goldwasser et al. (1988) and Krawczyk and Rabin (2000):

1. The two functions $f_0(x) = x^2 \mod n$, $f_1(x) = 4x^2 \mod n$, $x \in \{x \in \mathbb{Z}_n^* | x/p = x/q = 1\}$, are a pair of claw-free trapdoor permutations. Here,

$\{x \in \mathbb{Z}_n^* | x/p = x/q = 1\}$ is the set of quadratic residues modulo n.

2. With (p, q), the inverse permutations $f_0^{-1}(x) = x^{1/2} \mod n$, $f_1^{-1}(x) = (x/4)^{1/2} \mod n$, $x \in \{x \in \mathbb{Z}_n^* | x/p = x/q = 1\}$ can be efficiently computed.

3. Given one claw (x_0, x_1) such that $x_0^2 = 4x_1^2 \mod n$, trapdoor (p, q) can be extracted by $\gcd(x_0 \pm 2x_1, n)$.

Krawczyk and Rabin (2000) proposed one chameleon hash function based on the above claw-free trapdoor permutations $(f_0(\cdot), f_1(\cdot))$. This chameleon hash function plus the two algorithms (ChExp and ChIvt) which have been implicitly included in Krawczyk and Rabin (2000), is just the following invertible chameleon hash function with key exposure:

1. ChKg(1^k). Choose two distinct primes p, q such that $p = 3 \mod 8$, $q = 7 \mod 8$, and set inversion trapdoor itd $\leftarrow (p, q)$, collision trapdoor ctd $\leftarrow (p, q)$, and hash key chk $\leftarrow n$. Note that itd = ctd.

2. ChHash(chk, m, r). For chk $= n$, $r \leftarrow_R \mathbb{Z}_q^*$, parse the l-bit message $m = m_1, m_2, \ldots, m_l$, set initial value $h \leftarrow r^2 \mod n$, and then iteratively compute hash value $h \leftarrow 4^{m_i} h^2 \mod n$, for $i = 1, 2, \ldots, l$. Hence, we have ChHash(chk, m, r) $= f_{m_l}(\cdots(f_{m_2}(f_{m_1}(r^2)))\cdots) \mod n$.

3. ChCld(ctd, m, r, $\hat{m}$). Parse ctd $= (p, q)$ and compute $h \leftarrow$ ChHash(chk, m, r). For $i = l, l-1, \ldots, 1$, use (p, q) to iteratively compute value $h \leftarrow (h/4^{\hat{m}_i})^{1/2} \mod n$. At last, set $\hat{r} \leftarrow h^{1/2} \mod n$. Hence, it holds that $f_{\hat{m}_1}^{-1}(\cdots(f_{\hat{m}_{l-1}}^{-1}(f_{\hat{m}_l}^{-1}(\text{ChHash(chk, } m, r))))\cdots)^{1/2} =$ ChCld(ctd, m, r, $\hat{m}$).

4. ChExp(chk, m, r, $\hat{m}$, $\hat{r}$). Let h be the hash value and i^* the largest index such that $m_{i^*} \neq \hat{m}_{i^*}$. Set initial value $h \leftarrow r^2 \mod n$, $\hat{h} \leftarrow \hat{r}^2 \mod n$, and then iteratively compute $h \leftarrow 4^{m_i} h^2 \mod n$, $\hat{h} \leftarrow 4^{\hat{m}_i} \hat{h}^2 \mod n$, for $i = 1, 2, \ldots, i^* - 1$. Now it holds that $4^{m_{i^*}} h^2 = 4^{\hat{m}_{i^*}} \hat{h}^2$ and $m_{i^*} \neq \hat{m}_{i^*}$, $h \neq \hat{h}$. Hence, we obtain one claw (x_0, x_1) such that $x_0^2 = 4x_1^2 \mod n$, and then trapdoor (p, q) can be extracted by $\gcd(x_0 \pm 2x_1, n)$.

5. ChIvt(itd, h, m). Parse itd $= (p, q)$, and for $i = l, l-1, \ldots, 1$, use (p, q) to iteratively compute value $h \leftarrow (h/4^{m_i})^{1/2} \mod n$. At last, set $r \leftarrow h^{1/2}$. Hence, we have ChIvt(itd, h, m) $= f_{\hat{m}_1}^{-1}(\cdots(f_{\hat{m}_{l-1}}^{-1}(f_{\hat{m}_l}^{-1}(h)))\cdots)^{1/2}$.

Following the results in Krawczyk and Rabin (2000), the above chameleon hash function is provably collision resistant based on IF. Then by our generic DAPS construction, we immediately obtain the following DAPS scheme.

Remark 2 Here, we assume that there exists hash function H_1 with range $\text{QR}_n = \{x \in \mathbb{Z}_n^* | x/p = x/q = 1\}$. In fact, for RSA modulus n, it is widely believed that efficiently distinguishing elements in QR_n from elements in $\overline{\text{QR}}_n$ is a hard problem. However, as done in Poettering and Stebila (2014), by using the notion of sign-agnostic quadratic residues, the DAPS scheme using the group of sign-agnostic quadratic residues instead of the group of quadratic residues can be similarly constructed.

1. SKg(1^k). Choose two distinct primes p, q such that $p = 3 \mod 8$, $q = 7 \mod 8$, and set secret key sk $\leftarrow (p, q)$, and public key pk $\leftarrow n$.

2. SSign(pk, sk, sbj, msg). For sk $= (p, q)$, pk $= n$, $s \leftarrow_R \{0, 1\}^l$, set $h \leftarrow H_1(\text{sbj})$, $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$, compute $r = f_{m_1}^{-1}(\cdots(f_{m_{l-1}}^{-1}(f_{m_l}^{-1}(h)))\cdots)^{1/2} \mod n$, and set signature $\sigma \leftarrow (r, s)$.

3. SVrf(pk, sbj, msg, σ). For $\sigma = (r, s)$ and pk $= n$, if $H_1(\text{sbj}) = f_{m_l}(\cdots(f_{m_2}(f_{m_1}(r^2)))\cdots) \mod n$, then signature σ is valid; otherwise, σ is invalid.

4. SForge(pk, (sbj, msg, σ), (sbj, $\widehat{\text{msg}}$, $\hat{\sigma}$), (sbj*, msg*, σ^*), $\widehat{\text{msg}}^*$). For pk $= n$, $\sigma = (r, s)$, $\hat{\sigma} = (\hat{r}, \hat{s})$, $\sigma^* = (r^*, s^*)$, $\hat{s}^* \leftarrow_R \{0, 1\}^l$, set $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$, $\hat{m} \leftarrow H_2(\text{sbj}||\widehat{\text{msg}}||\hat{s})$, $\hat{m}^* \leftarrow H_2(\text{sbj}^*||\widehat{\text{msg}}^*||\hat{s}^*)$, and $h^* \leftarrow H_1(\text{sbj}^*)$, and compute itd $\leftarrow$ ChExp(chk, m, r, $\hat{m}$, $\hat{r}$), $\hat{r}^* \leftarrow$ ChIvt(ctd, h^*, $\hat{m}^*$), where itd = cdt $= (p, q)$. At last, set signature $\hat{\sigma}^* = (\hat{r}^*, \hat{s}^*)$.

Finally, it can be seen that the above DAPS scheme is double-signature forgeable in the trusted setup model, since the condition that $p = 3 \mod 8$, $q = 7 \mod 8$ is indispensable for algorithm ChExp.

6.2 DAPS based on RSA

First, we present the invertible chameleon hash function with key exposure based on RSA. Note that its variants or similar schemes have been widely mentioned, such as trapdoor commitment (Fischlin, 2001), the non-malleable commitment scheme (Fischlin and Fischlin, 2000), identity-based chameleon hash (Ateniese and de Medeiros, 2004a), multi-trapdoor commitment (Gennaro, 2004), exposure-free chameleon hash function (Ateniese and de Medeiros, 2004b), and Σ-hash functions (Bellare and Ristov, 2014).

1. ChKg(1^τ, 1^{l_m}). Choose two distinct primes p, q with τ-bit length and compute $n = pq$. Then choose another prime $e > 2^{l_m}$, where l_m is the parameter to ensure that e is large enough. Compute d such that $ed = 1 \mod (p-1)(q-1)$. Choose $x \leftarrow_R \mathbb{Z}_n^*$ and compute $y = x^e \mod n$. Set inversion trapdoor itd $\leftarrow d$, collision trapdoor ctd $\leftarrow x$, and hash key chk $\leftarrow (n, e, y)$.

2. ChHash(chk, m, r). For chk $= (n, e, y)$, compute value $h \leftarrow y^m r^e \mod n$, where $r \in \mathbb{Z}_n^*$, $m \in \{0, 1, \ldots, 2^{l_m} - 1\}$.

3. ChCld(ctd, m, r, $\hat{m}$). Parse ctd $= x$ and compute $\hat{r} = x^{m-\hat{m}} r \mod n$.

4. ChExp(chk, m, r, $\hat{m}$, $\hat{r}$). By $y^m r^e = y^{\hat{m}} \hat{r}^e \mod n$, we have $\hat{r}r^{-1} = x^{m-\hat{m}} \mod n$. By the extended Euclid algorithm, compute two integers $\bar{m}$, $\bar{e}$ such that $\bar{e}e + \bar{m}(m - \hat{m}) = 1$. Then compute $x \leftarrow y^{\bar{e}} (\hat{r}r^{-1})^{\bar{m}} \mod n$.

5. ChIvt(itd, h, m). For itd $= d$, compute $r \leftarrow (h/y^m)^d \mod n$.

Following the result on the chameleon hash function based on RSA (Bellare and Ristov, 2014), the above chameleon hash function is provably collision resistant based on RSA. By applying our generic DAPS construction, the following DAPS scheme is immediately constructed:

1. SKg(1^τ, 1^{l_m}). Choose two distinct primes p, q with τ-bit length and compute $n = pq$. Then choose another prime $e > 2^{l_m}$, and compute d such that $ed = 1 \mod (p-1)(q-1)$. Choose $x \leftarrow_R \mathbb{Z}_n^*$ and compute $y = x^e \mod n$. Set secret key sk $\leftarrow (x, d)$ and public key pk $\leftarrow (n, e, y)$.

2. SSign(pk, sk, sbj, msg). Parse sk $= (x, d)$, pk $= (n, e, y)$, randomly choose $s \leftarrow_R \{0,1\}^l$, and compute $h \leftarrow H_1(\text{sbj})$, $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$. Compute $r \leftarrow (h/y^m)^d \mod n$ and set signature $\sigma \leftarrow (r, s)$.

3. SVrf(pk, sbj, msg, σ). Parse $\sigma = (r, s)$ and pk $= (n, e, y)$. If $H_1(\text{sbj}) = y^m r^e \mod n$, where $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$, then signature σ is valid; otherwise, σ is invalid.

4. SForge(pk, (sbj, msg, σ), (sbj, $\widehat{\text{msg}}$, $\hat{\sigma}$), (sbj*, msg*, σ^*), $\widehat{\text{msg}}^*$). For pk $= (n, e, y)$, $\sigma = (r, s)$, $\hat{\sigma} = (\hat{r}, \hat{s})$, $\sigma^* = (r^*, s^*)$, $\hat{s}^* \leftarrow_R \{0,1\}^l$, set $h \leftarrow H_1(\text{sbj})$, $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$, $\hat{m} \leftarrow H_2(\text{sbj}||\widehat{\text{msg}}||\hat{s})$, $m^* \leftarrow H_2(\text{sbj}^*||\text{msg}^*||s^*)$, and $\hat{m}^* \leftarrow H_2(\text{sbj}^*||\widehat{\text{msg}}^*||\hat{s}^*)$. Compute $x \leftarrow y^{\bar{e}} (\hat{r}r^{-1})^{\bar{m}} \mod n$, $\hat{r}^* = x^{m^* - \hat{m}^*} r^* \mod n$, where $\bar{e}e + \bar{m}(m - \hat{m}) = 1$. At last, set signature $\hat{\sigma}^* = (\hat{r}^*, \hat{s}^*)$.

At last, recall that the above algorithm ChExp proceeds well for any public key (n, e, y) and any collision pair (m, r), $(\hat{m}, \hat{r})$, only if e is a prime and $(e, \phi(n)) = 1$ where $\phi(n) = |\{x | 1 < x < n, (x, n) = 1\}|$ is the Euler function of n. In fact, on the one hand, since the prime testing algorithm is well known in cryptography, the case that e is not a prime will not happen even in the untrusted setup model. On the other hand, if $(e, \phi(n)) \neq 1$ and e is large enough, then $\phi(n)$ and n will become significantly factorizable. Note that it is now required that $e > 2^{l_m}$. Hence, the above DAPS scheme is double-signature forgeable in the untrusted setup model.

6.3 DAPS based on CDH in the gap Diffie-Hellman group

Based on the chameleon hash function (Chen et al., 2011), we construct the invertible chameleon hash function with key exposure as follows:

1. ChKg(1^k). Choose the gap Diffie-Hellman (GDH) group $\mathbb{G} = \langle g \rangle$ with prime order q and generator g, where GDH (Chen et al., 2011) means that the computational Diffie-Hellman problem is hard, while the decisional Diffie-Hellman (DDH) problem is easy in this group. Then randomly choose $x \leftarrow_R \mathbb{Z}_q^*$ and set $y \leftarrow g^x$, inversion trapdoor itd $\leftarrow x$. Randomly choose $\bar{y} \leftarrow_R \mathbb{G}$ and compute collision trapdoor ctd $\leftarrow \bar{y}^x$. Lastly, set hash key chk $\leftarrow (g, y, \bar{y})$.

2. ChHash(chk, m, r). Randomly choose $a \leftarrow_R \mathbb{Z}_q$, compute $r \leftarrow (g^a, y^a)$, and compute hash value $h \leftarrow g^a \bar{y}^m$.

3. ChCld(ctd, m, r, $\hat{m}$). Parse $r = (g^a, y^a)$, ctd $= \bar{y}^x$, compute $g^{\hat{a}} \leftarrow g^a \bar{y}^{m-\hat{m}}$, $y^{\hat{a}} \leftarrow y^a \bar{y}^{x(m-\hat{m})}$, and set $\hat{r} \leftarrow (g^{\hat{a}}, y^{\hat{a}})$.

4. ChExp(chk, m, r, $\hat{m}$, $\hat{r}$). Parse $r = (g^a, y^a)$, $\hat{r} = (g^{\hat{a}}, y^{\hat{a}})$, chk $= (g, y, \bar{y})$, and compute collision trapdoor $\bar{y}^x \leftarrow (y^{\hat{a}}/y^a)^{(m-\hat{m})^{-1}}$.

5. ChIvt(itd, h, m). Parse itd $= x$, compute $g^a \leftarrow h/\bar{y}^m$, $y^a \leftarrow (g^a)^x$, and set $r \leftarrow (g^a, y^a)$.

Following the result (Chen et al., 2011), the above chameleon hash function is provably collision resistant. Then by our generic DAPS construction, we immediately obtain the following DAPS scheme:

1. SKg(1^k). Choose the GDH group $\mathbb{G} = \langle g \rangle$ with prime order q and generator g. Then randomly choose $x \leftarrow_R \mathbb{Z}_q^*$, $\bar{y} \leftarrow_R \mathbb{G}$. Set secret key sk $\leftarrow x$

and public key $y \leftarrow g^x$, $pk \leftarrow (g,\ y,\ \bar{y})$.

2. SSign(pk, sk, sbj, msg). Parse sk $= x$, pk $= (g,\ y,\ \bar{y})$, randomly choose $s \leftarrow_R \{0,1\}^l$, and compute $h \leftarrow H_1(\text{sbj})$, $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$. Compute $r \leftarrow (h/\bar{y}^m,\ (h/\bar{y}^m)^x)$ and set signature $\sigma \leftarrow (r,\ s)$.

3. SVrf(pk, sbj, msg, σ). Parse $\sigma = (r_1,\ r_2,\ s)$ and pk $= (g,\ y,\ \bar{y})$. If $(g,\ y,\ r_1,\ r_2)$ is a valid Diffie-Hellman tuple (i.e., $r_2 = r_1^x$) and $H_1(\text{sbj}) = r_1\bar{y}^{H_2(\text{sbj}||\text{msg}||s)}$, then signature σ is valid; otherwise, σ is invalid.

4. SForge(pk, (sbj, msg, σ), (sbj, $\widehat{\text{msg}}$, $\hat{\sigma}$), (sbj*, msg*, σ^*), $\widehat{\text{msg}}^*$). For pk $= (y,\ \bar{y})$, $\sigma = (g^a,\ y^a,\ s)$, $\hat{\sigma} = (g^{\hat{a}},\ y^{\hat{a}},\ \hat{s})$, $\sigma^* = (g^{a^*},\ y^{a^*},\ s^*)$, and $\hat{s}^* \leftarrow_R \{0,1\}^l$, set $h \leftarrow H_1(\text{sbj})$, $m \leftarrow H_2(\text{sbj}||\text{msg}||s)$, $\hat{m} \leftarrow H_2(\text{sbj}||\widehat{\text{msg}}||\hat{s})$, $m^* \leftarrow H_2(\text{sbj}^*||\text{msg}^*||s^*)$, $\hat{m}^* \leftarrow H_2(\text{sbj}^*||\widehat{\text{msg}}^*||\hat{s}^*)$, and compute $\bar{y}^x \leftarrow (y^{\hat{a}}/y^a)^{(m-\hat{m})^{-1}}$, $g^{\hat{a}^*} \leftarrow g^{a^*}\bar{y}^{m^*-\hat{m}^*}$, $y^{\hat{a}^*} \leftarrow y^{a^*}\bar{y}^{x(m^*-\hat{m}^*)}$. Finally, set signature $\hat{\sigma}^* = (g^{\hat{a}^*},\ y^{\hat{a}^*},\ \hat{s}^*)$.

At last, recall that the above algorithm ChExp proceeds well no matter how public key $(g,\ y,\ \bar{y})$ is arbitrarily generated. Hence, the above DAPS scheme is double-signature forgeable in the untrusted setup model.

7 Comparison

We present Table 1, which compares the efficiency and security of the above three DAPS schemes with that proposed in Poettering and Stebila (2014). For Table 1, we further present some comments:

1. For the DAPS scheme IF-DAPS, as shown in Remark 2, we consider its variant version which uses the group of sign-agnostic quadratic residues instead of the group of quadratic residues. Hence, the computation Jac is counted for the signing and verifying algorithms in Table 1.

2. PS-DAPS and IF-DAPS cannot ensure the DAP property in the untrusted setup model, unless the zero-knowledge techniques are involved (Poettering and Stebila, 2014), while RSA-DAPS and CDH-DAPS can directly ensure the DAP property in the untrusted setup model.

3. In Section 2, three algorithms that formalize the DAP property are compared. Although Forge is functionally weaker than SForge and Extract, in practice it is still strong enough to collapse the signing system. Additionally, it is the main reason why more efficient DAPS schemes such as RSA-DAPS and CDH-DAPS can be constructed.

4. Our generic DAPS scheme depends upon chameleon hash functions, while that in Poettering and Stebila (2014) is based on the so-called extractable 2 : 1 trapdoor functions. As a cryptographic primitive, a chameleon hash function (or trapdoor commitment) appears much more extensively in theory and practice than an extractable 2 : 1 trapdoor function. Hence, our generic framework is more general than the previous one.

8 Conclusions and future work

In this paper, we have revisited some basic issues concerning the recent notion of double-authentication-preventing signature. We have proposed a new formalization of the double-signature forgeability, which captures the DAP property with looser conditions. The property of invertibility was explicitly proposed for the chameleon hash function with key exposure. Then we have succeeded in constructing the new generic DAPS scheme based on an invertible chameleon hash function with key exposure. Three DAPS schemes based on the common assumptions of IF, RSA, and CDH, respectively,

Table 1 Comparison of different double-authentication-preventing signature schemes

Item	PS-DAPS (Poettering and Stebila, 2014)	IF-DAPS (Section 6.1)	RSA-DAPS (Section 6.2)	CDH-DAPS (Section 6.3)
Secret key	p, q	p, q	$d : 1 < d < \phi(n)$	$x : 1 < x < \lvert\langle\mathbb{G}\rangle\rvert$
Public key	$(n, t) : t \in \mathbb{Z}_n^*$	n	$(n, e, y) : y \in \mathbb{Z}_n^*, e > 2^l$	$(g, y, \bar{y}) : g, y, \bar{y} \in \mathbb{G}$
Signature	$(s, a_1, a_2 \ldots, a_l)$: $s, a_i \in \mathbb{Z}_n$	$(r, s) : r \in \mathbb{Z}_n$, $s \in \{0,1\}^l$	$(r, s) : r \in \mathbb{Z}_n$, $s \in \{0,1\}^l$	$(r_1, r_2, s) : r_i \in \mathbb{G}$, $s \in \{0,1\}^l$
Signing	$(l+1)(\text{Jac} + \text{Sqrt})$	$(l+1)(\text{Jac} + \text{Sqrt})$	2Exp	3Exp$'$
Verifying	$(l+1)(\text{Jac} + \text{Sqr})$	$(l+1)(\text{Jac} + \text{Sqr})$	2Exp	1Exp$'$ + 1DDH
Setup model	Trusted	Trusted	Untrusted	Untrusted
DAP (Section 2)	Extract	Extract	SForge	SForge

Jac: computation of Jacobi symbol modulo n; Sqrt: square root modulo n; Sqr: squaring modulo n; Exp: exponentiation modulo n; Exp$'$: exponentiation in group $\mathbb{G}$; DDH: verifying the decisional Diffie-Hellman tuple were instantiated. These results on DAPS are better than those in Poettering and Stebila (2014) in both theory and practice. Due to the rich cryptographic and algebraic properties of the new cryptographic primitive of invertible chameleon hash function, we expect to discover more applications for other relative cryptographic schemes (Chaum et al., 1988; Chor et al., 2000; Goyal, 2007; Mohassel, 2010; Fu et al., 2016).

References

Ateniese G, de Medeiros B, 2004a. Identity-based chameleon hash and applications. Int Conf on Financial Cryptography, p.164-180. https://doi.org/10.1007/978-3-540-27809-2_19

Ateniese G, de Medeiros B, 2004b. On the key exposure problem in chameleon hashes. Int Conf on Security in Communication Networks, p.165-179. https://doi.org/10.1007/978-3-540-30598-9_12

Bellare M, Ristov T, 2014. A characterization of chameleon hash functions and new, efficient designs. *J Cryptol*, 27(4):799-823. https://doi.org/10.1007/s00145-013-9155-8

Bellare M, Halevi S, Sahai A, et al., 1998. Many-to-one trapdoor functions and their relation to public-key cryptosystems. Annual Int Cryptology Conf, p.283-298. https://doi.org/10.1007/BFb0055735

Chaum D, Fiat A, Naor M, 1988. Untraceable electronic cash. Conf on the Theory and Application of Cryptography, p.319-327. https://doi.org/10.1007/0-387-34799-2_25

Chen X, Zhang F, Kim K, 2004. Chameleon hashing without key exposure. Int Conf on Information Security, p.87-98. https://doi.org/10.1007/978-3-540-30144-8_8

Chen X, Zhang F, Susilo W, et al., 2007. Efficient generic on-line/off-line signatures without key exposure. Int Conf on Applied Cryptography and Network Security, p.18-30. https://doi.org/10.1007/978-3-540-72738-5_2

Chen X, Zhang F, Tian H, et al., 2011. Discrete logarithm based chameleon hashing and signatures without key exposure. *Comput Electr Eng*, 37(4):614-623. https://doi.org/10.1016/j.compeleceng.2011.03.011

Chen X, Zhang F, Susilo W, et al., 2014. Identity-based chameleon hashing and signatures without key exposure. *Inform Sci*, 265(5):198-210. https://doi.org/10.1016/j.ins.2013.12.020

Chor B, Fiat A, Naor M, et al., 2000. Tracing traitors. *IEEE Trans Inform Theory*, 46(3):893-910. https://doi.org/10.1109/18.841169

Fischlin M, 2001. Trapdoor Commitment Schemes and Their Applications. PhD Thesis, Goethe Universitat Frankfurt, Germany.

Fischlin M, Fischlin R, 2000. Efficient non-malleable commitment schemes. Annual Int Cryptology Conf, p.413-431. https://doi.org/10.1007/3-540-44598-6_26

Fu Z, Ren K, Shu J, et al., 2016. Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE Trans Parall Distr Syst*, 27(9):2546-2559. https://doi.org/10.1109/TPDS.2015.2506573

Gao W, Wang X, Xie D, 2007. Chameleon hashes without key exposure based on factoring. *J Comput Sci Technol*, 22(1):109-113. https://doi.org/10.1007/s11390-007-9015-9

Gao W, Li F, Wang X, 2009. Chameleon hash without key exposure based on Schnorr signature. *Comput Stand Inter*, 31(2):282-285. https://doi.org/10.1016/j.csi.2007.12.001

Gennaro R, 2004. Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks. Annual Int Cryptology Conf, p.220-236. https://doi.org/10.1007/978-3-540-28628-8_14

Girault M, 1991. Self-certified public keys. Workshop on the Theory and Application of Cryptographic Techniques, p.490-497. https://doi.org/10.1007/3-540-46416-6_42

Goldwasser S, Micali S, Rivest R, 1988. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J Comput*, 17(2):281-308. https://doi.org/10.1137/0217017

Goyal V, 2007. Reducing trust in the PKG in identity based cryptosystems. Annual Int Cryptology Conf, p.430-447. https://doi.org/10.1007/978-3-540-74143-5_24

Krawczyk H, Rabin T, 2000. Chameleon signatures. 7th Network and Distributed System Security Conf, p.143-154.

Mohassel P, 2010. One-time signatures and chameleon hash functions. Int Workshop on Selected Areas in Cryptography, p.302-319. https://doi.org/10.1007/978-3-642-19574-7_21

Pedersen TP, Pfitzmann B, 1997. Fail-stop signatures. *SIAM J Comput*, 26(2):291-330. https://doi.org/10.1137/S009753979324557X

Poettering B, Stebila D, 2014. Double-authentication-preventing signatures. 19th European Symp on Research in Computer Security, p.436-453. https://doi.org/10.1007/978-3-319-11203-9_25

Poettering B, Stebila D, 2017. Double-authentication-preventing signatures. *Int J Inform Secur*, 16(1):1-22. https://doi.org/10.1007/s10207-015-0307-8

Soghoian C, Stamm S, 2011. Certified lies: detecting and defeating government interception attacks against SSL (short paper). Int Conf on Financial Cryptography and Data Security, p.250-259. https://doi.org/10.1007/978-3-642-27576-0_20