Digital signature schemes with strong existential unforgeability [version 1; peer review: awaiting peer review]

Jason Chia¹, Ji-Jian Chin², Sook-Chin Yip¹

¹Faculty of Engineering, Multimedia University, Cyberjaya, Selangor, 63100, Malaysia
²Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Selangor, 63100, Malaysia

Abstract
Digital signature schemes (DSS) are ubiquitously used for public authentication in the infrastructure of the internet, in addition to their use as a cryptographic tool to construct even more sophisticated schemes such as those that are identity-based. The security of DSS is analyzed through the existential unforgeability under chosen message attack (EUF-CMA) experiment which promises unforgeability of signatures on new messages even when the attacker has access to an arbitrary set of messages and their corresponding signatures. However, the EUF-CMA model does not account for attacks such as an attacker forging a different signature on an existing message, even though the attack could be devastating in the real world and constitutes a severe breach of the security system. Nonetheless, most of the DSS are not analyzed in this security model, which possibly makes them vulnerable to such an attack. In contrast, a better security notion known as strong EUF-CMA (sEUF-CMA) is designed to be resistant to such attacks. This review aims to identify DSS in the literature that are secure in the sEUF-CMA model. In addition, the article discusses the challenges and future directions of DSS. In our review, we consider the security of existing DSS that fit our criterion in the sEUF-CMA model; our criterion is simple as we only require the DSS to be at least secure against the minimum of existential forgery. Our findings are categorized into two classes: the direct and indirect classes of sEUF-CMA. The former is inherently sEUF-CMA without any modification while the latter requires some transformation. Our comprehensive review contributes to the security and cryptographic research community by discussing the efficiency and security of DSS that are sEUF-CMA, which aids in selecting robust DSS in future design considerations.

Keywords
Cryptography, Digital Signatures, Strong Existential Unforgeability
Corresponding authors: Jason Chia (1161300548@student.mmu.edu.my), Ji-Jian Chin (jjchin@mmu.edu.my), Sook-Chin Yip (scyip@mmu.edu.my)

Author roles: Chia J: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Chin JJ: Conceptualization, Funding Acquisition, Methodology, Project Administration, Resources, Supervision, Validation, Writing – Review & Editing; Yip SC: Methodology, Supervision, Validation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This research is supported by the Ministry of Higher Education of Malaysia through the Fundamental Research Grant Scheme under Grant FRGS/1/2019/ICT04/MMU/02/5 and in part by Multimedia University's Research Management Fund. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2021 Chia J et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Chia J, Chin JJ and Yip SC. Digital signature schemes with strong existential unforgeability [version 1; peer review: awaiting peer review] F1000Research 2021, 10:931 https://doi.org/10.12688/f1000research.72910.1

First published: 16 Sep 2021, 10:931 https://doi.org/10.12688/f1000research.72910.1
**Introduction**

The idea of a digital signature scheme (DSS) was proposed by Diffie and Hellman in 1976 as a necessity to design efficient authenticated electronic communications which can serve as legal evidence in the court of law. Rivest, Shamir, and Adleman realized the idea in their seminal work known as the RSA cryptosystem, the first of many. A DSS consists of three processes, as shown in Figure 1.

For every \((sk, pk) \leftarrow \text{keygen}(1^k)\) and every \(m\), verify\((\text{sign}(m, sk), m, pk) = 1\) must hold. A \(\sigma\) on \(m\) is valid if verify\((\sigma, m, pk) = 1\). This is a standard (informal) definition of DSS. In the early 90s, a paradigm known as hash-then-sign forms the industry standard for issuing digital signatures. The idea is to sign on the hash of a message, \(h \leftarrow H(m)\) instead of the \(m\) itself; this has a few benefits for \(h\) is constant size, which leads to efficient signing on speed and a constant sized signature. Verification would then require the verifiers to first validate the signed hash, then perform hashing \(h' \leftarrow H(m)\) before finally comparing \(h'\) and \(h\). Concrete examples of the hash-then-sign are discussed in.

**Properties of DSS**

The following properties are required by DSS:

- **Public verifiability**: A signature \(\sigma\) generated from a private key \(sk\) can be verified using a public key \(pk\). This property differentiates DSS from other integrity protection mechanisms, such as message authentication codes (MAC). A consequential result from this property is that signatures are also *transferrable*, meaning a party can copy \(\sigma\) and \(pk\) to use it to convince others that the message is authentic from the signer.

- **Non repudiation**: A signer cannot later deny that they have authenticated a message \(m\) once the signature of \(m, \sigma\) is generated and known. This is also another property that separates DSS from MAC, because the only entity that could have plausibly generated the signature in the case of DSS must possess the private key \(sk\); whereas in a MAC scheme, the keys are shared. This property also implies that only the one in possession of \(sk\) can generate valid signatures, which disallows forgeries.

<table>
<thead>
<tr>
<th>Input</th>
<th>Process</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>sec. parameter, (1^k)</td>
<td>keygen((1^k) \rightarrow (sk, pk))</td>
<td>public key, (pk)</td>
</tr>
<tr>
<td>message, (m)</td>
<td>sign((m, sk) \rightarrow \sigma)</td>
<td>signature, (\sigma)</td>
</tr>
<tr>
<td>private key, (sk)</td>
<td>verify((\sigma, m, pk) \rightarrow d)</td>
<td>validity, (d)</td>
</tr>
</tbody>
</table>

*Figure 1. The three processes of a digital signature scheme and the relationship between the input/outputs.*
Message recovery

Notice that the verification process requires both the message and the signature as inputs, requiring the signer to transmit both. Alternatively, some DSS can support message recovery. DSS with message recovery (DSS-R) has a different sign and verification process, shown in Figure 2; the signer only needs to transmit a packed signature \( \rho \), and the verifier would recover the message \( m \) successfully or abort \( \bot \) depending on signature validity. For practical purposes, \( |\rho| \leq |\sigma| + |m| \).

Examples of DSS-R are found in previous literature.16-23

Security models

The security of DSS was first formalized by Goldwasser, Micali and Rivest in 1988.24 Different security goals are used to model the different security guarantees of a DSS when faced with an adversary. The commonly accepted goal to model against is the goal of existential forgery (EUF), which is the easiest for attackers to achieve among other goals (e.g., selective forgery, total break). In addition,24 defined the adversarial capabilities for a DSS, which aims to model what types of attacks can be carried out by an adversary when attempting to break the DSS. The strongest capability, known as adaptive chosen message (CMA) is the widely accepted adversarial capability that is the most used in the literature on DSS. Figure 3 shows the interactions of a challenger and an adversary in the EUF-CMA model. In step (1), the challenger sets up an empty set \( Q \) and gives the \( pk \) to the adversary. In step (2), the adversary may make oracle queries that model its chosen message attack capability. The queried messages are added to the set \( Q \). In step (3), the adversary announces to the challenger the target message \( m^* \) it wants to forge. \( m^* \) must not be an element of \( Q \), nor it can be queried to the oracle; this prevents trivial attacks which uses the oracle to break the security goal. Note, step (4) allows the adversary to use the signing oracle again, which models the adaptive nature of the attack. Finally, in step (5), the adversary outputs a forgery...
σ*. We say the adversary breaks the DSS if σ* on m* is valid. qₖ quantifies the number message-signature pair made available to the adversary.

**Strong existential unforgeability (sEUF-CMA)**
Consider the case for randomized signatures in the EUF-CMA model (e.g., DSA¹¹ or PSS-R¹³). A randomized DSS allows multiple valid signatures for a single message, which has a subtle implication on the model: Suppose that the adversary queried for a message-signature pair (m, σ). Now, the adversary forges a different valid signature σ₀ on the same message m. This is an easier security goal, but could be a critical vulnerability when the DSS is used in a scenario in which the designers assumed that no new signatures can be forged, because a different signature on an existing message is still new. In other words, EUF-CMA does not guarantee that if an attacker knows (m, σ), it cannot forge (m, σ₀) such that σ₀ on m is valid. This gave rise to a stronger security model, known as strong existential unforgeability or sEUF-CMA.²⁵ Figure 4 shows the interactions of the adversary with the challenger in the sEUF-CMA model. The main difference is the constraint during the chosen message attacks and in the final step (5). Notably, the adversary can even query for signatures on the challenge message m*, but may not submit any of the signatures obtained from the sign oracle as forgeries. In contrast to EUF-CMA, sEUF-CMA ensures that an adversary cannot produce any new signatures at all; any valid signature must have originated from the signer.

**Why does sEUF-CMA matter?**
Let’s take a step back and consider why sEUF-CMA is even worth considering in the first place. DSS is very often used as a building block to construct other cryptographic schemes. For example, using the Fiat-Shamir transform,²⁶ a DSS can be turned into an identification protocol. DSS has also been used to create identity-based identification schemes,²⁷ signcryption schemes,²⁵ authenticated key exchanges,²⁸ and identity-based encryption schemes.²⁹ DSS with EUF-CMA security is found to be insufficient in some of the constructions, particularly to build non-malleable cryptographic schemes. In a nutshell, non-malleability refers to the impossibility of an adversary to generate a different ciphertext to some previously known ciphertext that decrypts to the same message, which is a desirable property in cryptographic schemes.³⁰ Thus, if a DSS is sEUF-CMA, it is much more versatile because it can be useful as a building block for many of the schemes which requires the property of non-malleability.

**A simple attack for when a DSS is not sEUF-CMA**
We show a toy example of a simple attack that can be achieved by an attacker if the DSS used for authentication is not sEUF-CMA. Figure 5 shows honest users Alice and Bob, as well as an attacker Mallory which has hijacked the channel. In step (1), Bob wants to authenticate that Alice is truly on the other end. Mallory launches a chosen message attack in step (2) and (3) on Alice and obtains the message m and signature σ. In step (4), Mallory forwards (m, σ) to Bob, which may initially convince them. After some time elapsed, Bob wants to re-authenticate Alice to ensure that they are still who they claim to be (5). Mallory forges a valid signature σ₀ ≠ σ on m and forwards that to Bob (6). From Bob’s perspective, Alice generating a different signature ought to warrant some confidence that Alice is truly Alice. However, if the DSS is not sEUF, this is not the case, as Bob is obviously duped.
Real life problems arising from signature malleability (non-sEUF)
We established sEUF-CMA is of theoretical interest. Here we briefly mention a few real-life needs for sEUF-CMA to emphasize the importance of it. Not too long ago, Stern et al. showed how to duplicate ECDSA signatures based on their malleability. A more recent example is the transaction malleability of the popular cryptocurrency, Bitcoin. When a transaction in Bitcoin is signed, the signature does not cover the entire transaction used for hashing; an attacker could perform strong forgery (i.e., maul the signature), then claim that the transaction has failed (because the hash is not valid). The sender would believe it because the signature is valid! This leads to the sender issuing a new transaction, without knowing that the original transaction is valid. Decker and Wattenhofer pointed out that this subtle flaw could be responsible for a small portion of Bitcoins stolen during the 2014 MtGox attack. They remarked that transaction malleability should be carefully considered when implementing Bitcoin clients. In another research, Jackson et al. found that signatures that are not sEUF-CMA may cause the strong session agreement of a well-known authenticated key exchange algorithm, the STS-ISO to fail. They analyzed STS-ISO using the Tamarin Prover and found that if sEUF is violated, attackers can force the parties in the session to accept message that do not originate from either of the honest parties.

DSS secure in sEUF-CMA
The notion of sEUF-CMA first appeared in the form of non-malleability of ciphertexts in the works of Dolev, Dwork and Naor in 1991. The term sEUF-CMA was first introduced by An, Dodis and Rabin which presented it as requirement to construct signcryption schemes. Boneh, Shen and Waters were the first DSS that is considered and proven in the sEUF-CMA model, but mentions that DSS that are sEUF-CMA have existed earlier such as full domain hash (FDH)-RSA. FDH-RSA must be sEUF-CMA because the signatures generated are determined exactly by the message only. This means that FDH-RSA signatures are deterministic. For deterministic signatures, a strong forgery cannot occur without the message $m'$ being different from the original $m$, therefore, deterministic signatures are generally sEUF-CMA. We see that this is also the case for the widely used and popular EdDSA which had recently received a provable security treatment. Boneh, Shen and Waters shows that non-deterministic signatures may also exhibit sEUF-CMA, such as the Micali-Reyzin signatures, Goh-Jarecki signatures and Boneh-Boyen signatures. The reason for this is that if the forger manages to re-randomize a signature on the same message, the signature constitutes an existential forgery through clever binding of the messages. This is a recurring paradigm to design sEUF-CMA signature schemes. Since the introduction of the sEUF-CMA model, existing EUF-CMA secure DSS are now re-considered in the sEUF-CMA model. An example of such work is by Fersch, Kiltz and Poettering on the well established DSA. DSA was shown to be secure under sEUF-CMA, while ECDSA was not. More recently in 2021, Brendel et al. proved the IETF version of EdDSA to be sEUF-CMA secure. Table 1 shows a list of DSS that are secure under sEUF-CMA.

---

1STS stands for Station-to-Station and is an authenticated key exchange protocol due to Diffie et al.
2Brown proved ECDSA to secure under sEUF-CMA, but this was refuted through a series of trivial attacks posed by.
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Req.</th>
<th>Sec. Assump</th>
<th>Sign Cost</th>
<th>Ver. Cost</th>
<th>Sig. len.</th>
<th>M.R.</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lamport</td>
<td>N/A</td>
<td>1-way functions</td>
<td></td>
<td>1k</td>
<td>hash/0</td>
<td>1k</td>
<td>hash</td>
</tr>
<tr>
<td>DSA</td>
<td>random oracle</td>
<td>DLP</td>
<td>0/ex</td>
<td>ex</td>
<td></td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>FDH-RSA</td>
<td>random oracle</td>
<td>RSA</td>
<td>0/ex</td>
<td>ex</td>
<td></td>
<td></td>
<td>(</td>
</tr>
<tr>
<td>CS99</td>
<td>N/A</td>
<td>Strong RSA</td>
<td>0/4ex</td>
<td>4ex</td>
<td></td>
<td>3</td>
<td>(</td>
</tr>
<tr>
<td>BLS01</td>
<td>random oracle</td>
<td>GDH</td>
<td>0/ex</td>
<td>pair</td>
<td></td>
<td></td>
<td>(</td>
</tr>
<tr>
<td>BMS03</td>
<td>N/A</td>
<td>CDH</td>
<td>( \log_2(n)^{\text{pair}} + \text{ex} ) ( \log_2(n)^{\text{ex}} + 1 )</td>
<td>( \log_2(n)^{\text{pair}} )</td>
<td></td>
<td></td>
<td>(</td>
</tr>
<tr>
<td>Gj03</td>
<td>random oracle</td>
<td>DDH</td>
<td>0/2ex</td>
<td>2ex</td>
<td></td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>KW03</td>
<td>random oracle</td>
<td>RSA</td>
<td>0/ex</td>
<td>ex</td>
<td></td>
<td>(</td>
<td>Z_n</td>
</tr>
<tr>
<td>BB04</td>
<td>N/A</td>
<td>Strong DH</td>
<td>0/ex</td>
<td>pair</td>
<td></td>
<td></td>
<td>(</td>
</tr>
<tr>
<td>mNR04</td>
<td>N/A</td>
<td>GGM</td>
<td>0/ex</td>
<td></td>
<td></td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>GJK +07</td>
<td>random oracle</td>
<td>CDH</td>
<td>0/3ex</td>
<td>2ex</td>
<td></td>
<td>( \log_2(n)^{\text{pair}} + 2</td>
<td>Z_q</td>
</tr>
<tr>
<td>GJK +07</td>
<td>random oracle</td>
<td>DDH</td>
<td>2ex/0</td>
<td>2ex</td>
<td></td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>TP09</td>
<td>random oracle</td>
<td>RSA &amp; DLP</td>
<td>0/2ex</td>
<td>3ex</td>
<td></td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>AGH +11</td>
<td>N/A</td>
<td>GGM</td>
<td>0/(2+kmn)^{\text{ex}}</td>
<td>(kmn)^{\text{pair}}</td>
<td></td>
<td>3</td>
<td>(</td>
</tr>
<tr>
<td>EdDSA</td>
<td>random oracle</td>
<td>ECDLP</td>
<td>0/2ex</td>
<td>2ex</td>
<td></td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>NTC19</td>
<td>random oracle</td>
<td>1-way trapdoor sampleable relations</td>
<td>0/2ex</td>
<td>2ex</td>
<td></td>
<td>(</td>
<td>G</td>
</tr>
</tbody>
</table>

**Table 1. DSS that are directly sEUF-CMA.**

**Legend:**
- **Req.** - Requirements
- **Sec. Assump** - Security Assumptions
- **Sign Cost** - Partitioned into off-line/on-line costs, separated by a slash (/);
- **Ver. Cost** - Verification Cost
- **Sig. Len.** - Signature Length
- **M.R.** - Supports message recovery
- **Std.** - Standard Model as oppose to random oracle model

- **DLP** - Discrete Logarithm Problem
- **RSA** - RSA problem
- **GDH** - Gap Diffie-Hellman
- **CDH** - Computational Diffie-Hellman
- **ECDLP** - Elliptic Curve Discrete Logarithm Problem
- **GGM** - Generic Group Model
- **DDH** - Decisional Diffie-Hellman Problem
- **ex** - Exponentiation (or equivalent to point multiplication in elliptic curves)
- **pair** - Pairing operation
- **hash** - Hash operation
- **G** - Group element
- \( | Z_p | \) - Integer of order \( p \) (a scalar in elliptic curves)
- \( | Z_n | \) - Integer of order \( n \) (a scalar in elliptic curves)
- \( G_x \) - X coordinate of elliptic curve point
- \( nm \) - Total number of messages
- \( kmn \) - Key matrix size (see 52)

Costs ignore cheap operations such as integer multiplication/addition or elliptic curve point addition/doubling. Whenever possible, the Strauss-Shamir optimization is applied (see 54).
Table 2. Conversions that produces DSS with sEUF-CMA.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>DSS Req.</th>
<th>Add. Req.</th>
<th>Sec. Assump</th>
<th>Sign Cost</th>
<th>Ver. Cost</th>
<th>Sig. len.</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR00</td>
<td>factoring-based Fiat-Shamir</td>
<td>random oracle</td>
<td>integer fact.</td>
<td>0/2ex</td>
<td>2ex</td>
<td>2</td>
<td>Zq</td>
</tr>
<tr>
<td>BSW06</td>
<td>EUF-CMA, partitioned</td>
<td>randomized trapdoor</td>
<td>CDH + trapdoor</td>
<td>0/sg + ex + td</td>
<td>vf + ex + td</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOO06-1</td>
<td>EUF-CMA</td>
<td>random oracle</td>
<td>DLP</td>
<td>sg + ex</td>
<td>0</td>
<td>vf + ex</td>
<td></td>
</tr>
<tr>
<td>TOO06-2</td>
<td>EUF-CMA</td>
<td>collision resistant hash</td>
<td>DLP + collision resist</td>
<td>sg + ex</td>
<td>0</td>
<td>vf + ex</td>
<td></td>
</tr>
<tr>
<td>SPW07</td>
<td>EUF-CMA</td>
<td>strong randomized trapdoor</td>
<td>trapdoor</td>
<td>0/sg + 2td</td>
<td>vf + 2td</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BS07</td>
<td>EUF-CMA</td>
<td>2-tier DSS</td>
<td>EUF-CMA of 2-tier DSS</td>
<td>0/2sg + kg</td>
<td>2vf</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Goldreich</td>
<td>EUF-CMA</td>
<td>1-time DSS</td>
<td>strong 1-time EUF DSS</td>
<td>sg(log₂nm)/sg</td>
<td>vf(log₂nm)(log₂nm) + 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HWZ07</td>
<td>EUF-CMA</td>
<td>1-time DSS</td>
<td>strong 1-time EUF DSS</td>
<td>sg + kg/sig</td>
<td>2vf</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>LKZ + 08-S</td>
<td>EUF-GMA, deterministic</td>
<td>N/A</td>
<td>N/A</td>
<td>sg + kg/sig</td>
<td>2vf</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>LKZ + 08-P</td>
<td>EUF-GMA, deterministic</td>
<td>N/A</td>
<td>N/A</td>
<td>sg/sig</td>
<td>2vf</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>LAS + 10</td>
<td>EUF-GMA/CMA</td>
<td>GTOW chameleon hash</td>
<td>one-wayness of GTOW hash</td>
<td>sg/td</td>
<td>vf + td</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DSS with leakage resilience under sEUF-CMA**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>DSS Req.</th>
<th>Add. Req.</th>
<th>Sec. Assump</th>
<th>Sign Cost</th>
<th>Ver. Cost</th>
<th>Sig. len.</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>KV09-2,1</td>
<td>Lamport DSS</td>
<td>1-time use 1-way functions</td>
<td>1k</td>
<td>hash/0</td>
<td>1k</td>
<td>hash</td>
<td>m</td>
</tr>
<tr>
<td>WT14</td>
<td>EUF-FLR</td>
<td>strong randomized trapdoor</td>
<td>trapdoor leakage resilience</td>
<td>0/sg + 2td</td>
<td>vf + 2td</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WT15</td>
<td>EUF-FLR</td>
<td>strong randomized trapdoor</td>
<td>trapdoor leakage resilience</td>
<td>0/sg + 2td + pair + ex</td>
<td>vf + 2td</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HHP16</td>
<td>EUF-FLR</td>
<td>1-time EUF-FLR DSS</td>
<td>strong 1-time EUF-FLR DSS</td>
<td>sg + kg/sig</td>
<td>2vf</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- DSS Req. - Requirement for DSS before using the conversion; Add. Req. - Additional requirements; EUF-GMA - Existential Unforgeability under Generic Chosen Message Attack; GTOW - Given Target One-Wayness (See 60); EUF-FLR - Existential Unforgeability with Full Leakage Resilience; sg - signature generation cost; td - trapdoor or operation cost; kg - key generation cost; 1k - security parameter; vf - signature verification cost; σ - underlying signature length; pk underlying signature public key length; π - Groth-Sahai proof statement (See 66).
Message recovery
From another perspective, we consider DSS with message recovery (DSS-R) candidates under sEUF-CMA. As pointed out by Ateniese and de Mediros, the modified Nyberg-Rueppel signature is sEUF-CMA secure. While it is tempting to think that DSS-R must be sEUF-CMA, because modifying the signature will surely modify the message itself given that one can recover the message from the signature. However, we see that this is not the case as Ateniese and de Mediros found the original Nyberg-Rueppel signatures to be insecure in sEUF-CMA.

Conversions to sEUF-CMA
Boneh, Shen and Waters opened a new field in the research of DSS under sEUF-CMA: Interest in sEUF-CMA conversion starts to accumulate with various works being published. Instead of working on individual DSS, the line of research focuses on creating efficient conversions to enhance existing generic DSS with EUF-CMA into sEUF-CMA. Some transforms do not even need EUF-CMA security at minimum, only requiring EUF-GMA, which is a weaker notion than EUF-CMA. Table 2 shows a list of conversion methods since 2006, including the use of DSS in leakage resilient settings.

Widely used DSS that have malleable signatures
Through our research, we investigated some of the most used DSS in the industry on their security in the sEUF-CMA model. The following are the popular DSS that are not sEUF-CMA secure.

1. RSA PKCS#1.5
2. ECDSA
3. Ed25519 (Original, not IETF RFC 8032)

Challenges and future direction
In recent work, there has been several post-quantum cryptographic DSS that incorporates the sEUF-CMA model during design. DSS in various other contexts (e.g., privacy preserving computation, multiparty computation) such as a homomorphic DSS, group DSS, and proxy DSS are being considered in sEUF-CMA as well. In addition, DSS in even more complex cryptographic settings such as in certificateless and identity-based settings are also using sEUF-CMA as their standard model for security. We see that the security goal post has been moved from EUF-CMA to sEUF-CMA in the span of a decade and believe this is the right direction forward as DSS is increasingly used in intricate security protocols, which cannot tolerate any design flaw that arises from as simple as malleable signatures.

Conclusion
In this work, we provided a comprehensive review on what is strong unforgeability in DSS, why is it needed, which of the DSS are secure under the model and how to obtain it if the DSS is only existentially unforgeable. We surveyed and analyzed existing DSS in literature which are secure under sEUF-CMA, and noted the requirements, computational and storage efficiency as well as the security assumptions of each DSS to provide an overview of DSS under the much more secure model.

Data availability
No data is associated with this article.

Acknowledgements
We would like to acknowledge our research funder: This research is supported by the Ministry of Higher Education of Malaysia through the Fundamental Research Grant Scheme under Grant FRGS/1/2019/ICT04/MMU/02/5 and in part by Multimedia University’s Research Management Fund.

References


Publisher Full Text


Publisher Full Text


Publisher Full Text


Publisher Full Text


Publisher Full Text


Publisher Full Text | Reference Source


Publisher Full Text


Publisher Full Text


The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com