# ECS227 Paper Writeup
# *Partial Signatures and their Applications* [1]

Ryan Stevens

11 March 2012

## Goals

The goal of this paper is to provide a digital signature scheme that allows the signer of a message to preserve their anonymity until they want to claim that the message was really sent by them. Thus it becomes up to the signer when the message can be verified and the identity of the signer known. Such a scheme has applications in anonymous bidding, for example, where a bidder wishes to place a bid but wants to remain anonymous until they know they have won the auction. The authors lay out three goals for such a scheme: anonymity, unambiguity, and unforgeability. This differs from the goals of traditional public-key digital signature schemes in that these schemes do not guarantee the anonymity of the signer; in fact traditional digital signatures are designed to make it easy to verify the identity of the signer assuming the public key of the signer is known.

## Partial Signatures

To achieve these goals, the authors outline a type of digital signature scheme called *partial signatures* in which the signature consists of two parts: a "stub" $\sigma$ and a "de-anonymizer" $\kappa$. The stub is sent with the message to be verified and is insufficient to verify the identity of the signer. Only when the signer provides the de-anonymizer can the public key of the signer be used to verify the message, and thus the identity of the signer known. If Alice wishes to use such a partial signature scheme to sign a message anonymously, she must generate a secret key $sk$ and a public key $vk$ (verification key) just like a traditional public key digital signature scheme. With the secret key and the message, Alice produces the signature $(\sigma, \kappa)$ and sends $(M, \sigma)$ to the receivers, who cannot verify Alice's identity with $\sigma$ and $M$ alone. If she wishes to reveal her identity, she can send $\kappa$ to the receivers who can use $vk$ to verify the message really was from Alice.

The authors provide constructions for creating a partial signature scheme from a digital signature scheme and a commitment scheme. A commitment scheme allows a sender to commit to a decision but hide what decision they made until they want to reveal it. Before we go into detail, digital and partial signature schemes must be defined as well as commitment schemes:

- *Digital Signature Scheme*: A digital signature scheme $\mathcal{DS}$ is a triple of algorithms $(\mathsf{SKG}, \mathsf{SIG}, \mathsf{SVF})$. $\mathsf{SKG}$ is the key generation algorithm that produces the $(sk, vk)$ pair of keys. $\mathsf{SIG}$ is the signature generation algorithm that takes as input a message $M$ and secret key $sk$ and produces the signature $s$. Lastly, $\mathsf{SVF}$ takes a message $M$, signature $s$, and public key $vk$ and outputs a boolean whether the message and secret key were used to produce $s$.

- *Partial Signature Scheme*: A partial signature scheme $\mathcal{PS}$ is a triple of algorithms $(\mathsf{PKG}, \mathsf{PSIG}, \mathsf{PVF})$ that is similar to a digital signature scheme, except $\mathsf{PSIG}$ now outputs $s = (\sigma, \kappa)$.

- *Commitment Scheme*: A commitment scheme $\mathcal{CMT}$ is a pair of algorithms $(\mathsf{CMT}, \mathsf{CVF})$. The commitment algorithm $\mathsf{CMT}$ takes a message $M$ as input and returns the commitment value $\sigma$ and the decommital value $\omega$. The verification algorithm $\mathsf{CVF}$ takes $(\sigma, \omega, M)$ as input and returns a boolean that reveals if the tuple is valid.

The three security notions the authors outline for secure partial signature scheme are: anonymity, unambiguity, and unforgability. These are defined as follows:

- *Anonymity*: The identity of the signer should not be able to be determined from the tuple $(M, \sigma)$ alone. What this means is that assuming the adversary has access to a number of verification keys (at worst two), then they should not be able to determine which of the verification keys should be used to verify $(M, \sigma)$ until $\kappa$ is known.

$$\mathbf{Adv}^{an}_{\mathcal{PS}}(A) = 2 * \mathbf{Pr}[b \overset{\$}{\leftarrow} \{0,1\}; (vk_0, sk_0, vk_1, sk_1) \overset{\$}{\leftarrow} \mathsf{PKG}();$$
$$A^{\mathsf{PSIG}(\bullet, sk_b)}(vk_0, sk_0, vk_1, sk_1) \rightarrow d : d = b] - 1$$

- *Unambiguity*: The stub $\sigma$ needs to be linked with the signer's secret key in an unambiguous way. The authors point out that using the empty string as $\sigma$ and the full signature as $\kappa$ ensures anonymity, but an adversary could easily produce a valid $\kappa$ under *their* verification key, leading to the adversary claiming ownership of the original message. Thus, $\sigma$ must be tied to the signers private key. This notion is defined in the following way: the adversary picks two distinct verification keys $vk_0$ and $vk_1$, any two messages $M_0$ and $M_1$, and two de-anonymizers $\kappa_0$ and $\kappa_1$. They win if they are able to produce a singe $\sigma$ that is valid for both $(vk_0, M_0, \kappa_0)$ and $(vk_1, M_1, \kappa_1)$.

$$\mathbf{Adv}^{ua}_{\mathcal{PS}}(A) = \mathbf{Pr}[A() \rightarrow (vk_0, vk_1, M_0, M_1, \sigma, \kappa_0, \kappa_1) : \mathsf{PVF}(vk_0, M_0, \sigma, \kappa_0) = 1 \wedge$$
$$\mathsf{PVF}(vk_1, M_1, \sigma, \kappa_1) = 1 \wedge vk_0 \neq vk_1]$$

- *Unforgeability*: In the above definition, the adversary was able to use its own verification key instead of the signer's to forge a message. This differs from traditional unforgability in that the adversary should not be able to forge a message under the signer's verification key; that is, given the message and stub, the adversary cannot compute the de-anonymizer. The notion here is defined as follows: the adversary can query an oracle $O_\sigma$ that gives a stub for some input message and an oracle $O_\kappa$ that gives the de-anonymizer for some stub. If the adversary can provide a de-anonymizer for a stub that it got from $O_\sigma$ which it did not use as input to, $O_\kappa$, than it wins.

$$\mathbf{Adv}^{uf}_{\mathcal{PS}}(A) = \Pr[(vk, sk) \xleftarrow{\$} \mathsf{PKG}(); A^{O_\sigma(\bullet), O_\kappa(\bullet)}(vk) \to (M, \sigma, \kappa) : \\ \mathsf{PVF}(vk, M, \sigma, \kappa) = 1 \wedge A \text{ did not ask } O_\kappa(\sigma)]$$

Now we go over the simplest of the authors' constructions of partial signature schemes.

## Designing Partial Signature Schemes

The authors propose a partial signature scheme called Sign-then-Commit (StC) that leverages both a commitment scheme and a digital signature scheme. Assuming we have $\mathcal{DS} = (\mathsf{SKG}, \mathsf{SIG}, \mathsf{SVF})$ and $\mathcal{CMT} = (\mathsf{CMT}, \mathsf{CVF})$ we construct $\mathcal{PS} = (\mathsf{PKG}, \mathsf{PSIG}, \mathsf{PVF})$ in the following way:

```
Alg PKG():
$(vk, sk) \leftarrow \mathsf{SKG}()$
return$(vk, sk)$
```

```
Alg PSIG$(sk, M)$:
$s \xleftarrow{\$} \mathsf{SIG}(sk, M)$
$(\sigma, \omega) \xleftarrow{\$} \mathsf{CMT}(s||vk)$
$\kappa \leftarrow (s, \omega)$
return$(\sigma)$
```

```
Alg PVF$(vk, M, \sigma, \kappa)$:
$(s, \omega) \leftarrow \kappa)$
if $\mathsf{CVT}(\sigma, s||vk, \omega) = 1$ then
    if $\mathsf{SVF}(vk, s, M) = 1$ then
        return(1)
return(0)
```

We use the key generation algorithm from the digital signature scheme directly as our key generation algorithm. To make the stub and de-anonymizer, we use the stub generation scheme from the commitment scheme and give it the signature of our message concatenated with the verification key. Lastly to verify the message, we check that de-anonymizer is the correct commit value for the commitment scheme and that the signature is the correct signature for the message and public key.

The authors' proof of the unforgeability notion of StC is quite complicated and I could only understand it very loosely, so I will not attempt to recreate the proof here. The proofs of anonymity and unambiguity are easier to understand and I go over both below, although they are very similar.

**Proof of Anonymity**

The proof that the StC achieves the anonymity security notion goes as follows:

$$\begin{gathered}\mathcal{CMT} \text{ is hiding} \rightarrow \mathcal{PS} \text{ is anonymous}\\ \text{Adversary } A \text{ gets good advantage against } \mathcal{PS} \text{ anonymity security} \rightarrow\\ \text{Adversary } B \text{ gets good advantage against } \mathcal{CMT} \text{ hiding security}\end{gathered}$$

To continue, we must define the $\mathcal{CMT}$ hiding security notion. A commitment scheme is good at the hiding security notion if an adversary is not able to determine whether a commitment value $\sigma$ belongs to a particluar message. Stated formally:

$$\mathbf{Adv}^{hd}_{\mathcal{CMT}}(A) = Pr[b \overset{\$}{\leftarrow} \{0,1\}; [A^{\mathsf{CMT}(\mathbf{lr}_b(\bullet,\bullet))}() \rightarrow d : d = b]$$

Now, we design the adversary $B$ that breaks hiding security when given an adversary $A$ that breaks partial signature anonymity:

```
$(vk_0, sk_0, vk_1, sk_1) \overset{\$}{\leftarrow} \mathsf{CMT}()$
Run $A(vk_0, sk_0, vk_1, sk_1)$
When $A$ queries for the stub of $M$
    $s_0 \overset{\$}{\leftarrow} \mathsf{SIG}(sk_0, M)$
    $s_1 \overset{\$}{\leftarrow} \mathsf{SIG}(sk_1, M)$
    Query $B$'s $\mathsf{CMT}$ left-right oracle with parameters $s_0||vk_0$ and $s_1||vk_1$ to get $\sigma$
    Give $\sigma$ to $A$
When $A$ outputs its guess $d$, return $d$
```

From this we conclude that the probability that $B$ outputs that it is in the "left" oracle world is exactly the probability that $A$ is able to determine it is in the "left" oracle world. Thus, if $A$ is good at distinguishing which message the stub belongs to for partial signatures, $B$ will be able to do so for commitment schemes as well.

**Proof of Unambiguity**

The proof that the StC achieves the unambiguity security notion goes as follows:

$$\begin{gathered}\mathcal{CMT} \text{ is binding} \rightarrow \mathcal{PS} \text{ is unambiguous}\\ \text{Adversary } A \text{ gets good advantage against } \mathcal{PS} \text{ unambiguity security} \rightarrow\\ \text{Adversary } B \text{ gets good advantage against } \mathcal{CMT} \text{ binding security}\end{gathered}$$

To continue, we must define the $\mathcal{CMT}$ binding security notion. A commitment scheme is good at the binding security notion if an adversary is able to produce a pair of messages and a pair of decommit values that are both valid for a single stub. Stated formally:

$$\mathbf{Adv}^{bnd}_{\mathcal{CMT}}(A) = Pr[A() \to (M_0, M_1, \omega_0, \omega_1, \sigma) :$$
$$\mathsf{CVF}(M_0, \omega_0, \sigma) = 1 \land \mathsf{CVF}(M_1, \omega_1, \sigma) = 1 \land M_0 \neq M_1]$$

Now, we design the adversary $B$ that breaks binding security when given an adversary $A$ that breaks partial signature unambiguity:

```
Run $A()$
When $A$ outputs $(vk_0, vk_1, M_0, M_1, \sigma, \kappa_0, \kappa_1)$, return $(s_0||vk_0, s_1||vk_1, \omega_0, \omega_1, \sigma)$
```

It is easy to see that because of how StC is constructed, the value $B$ returns will always be valid for the commitment scheme if $A$ returns a correct tuple.

## Concluding Thoughts

I have shown how and why the authors chose to design partial signature schemes. They laid out a number of security notions that are appropriate to partial signatures and how these notions are defined formally. Lastly, they describe how to build partial signature schemes from existing digital signature schemes, which is covered at a high level in this writeup.

The remaining parts in the paper I did not cover involve how to build a partial signature scheme from a digital signature algorithm directly and how to build a commitment scheme from a partial signature scheme. The authors use a protocol called Schnorr [2] and modify it slightly to produce a very fast partial signature scheme. Since we did not cover how Schnorr operates in class, I chose to ignore the part of the paper that deals with this protocol. The section that describes how to make a commitment scheme from a partial signature scheme is straightforward, but since I did not cover how to build a partial signature scheme without a commitment scheme, this construction seemed pointless. Lastly, there is an analysis of the performance of each scheme the authors proposed as well as a more efficient version of the scheme I described in this writeup, however I didn't think it would add much to include it here.

## References

[1] M. Bellare and S. Duan. Partial signatures and their applications, 2009.

[2] C.P. Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.