6.889 New Developments in Cryptography April 12, 2011

Leakage-Resilient Digital Signatures

Instructors: Shafi Goldwasser, Yael Kalai, Leo Reyzin, Boaz Barak, and Salil Vadhan

Lecturer: Yael Kalai *Scribe: Jonathan Ullman*

In this class we gave a definition of leakage-resilient signatures in the bounded memory leakage model and then gave a construction due to Katz and Vaikuntanathan [KV09].

1 Review of Leakage Models

Before constructing leakage-resilient signatures we will review and clarify the different leakage models we've seen in class. Loosely, they fall into the following categories:

- **Computational Leakage** (aka the "Only Computation Leaks" Model) [MR04]: In this model we assume there is trusted storage and the adversary can only obtain leakage from data used in computation. The focus is usually on *continual leakage*, where the total leakage is unbounded and we use new randomness to periodically refresh the key.
- **Memory Leakage** [AGV09]: In this model the adversary can obtain leakage from the entire secret key and sometimes also from the entire internal state of the system. Memory leakage comes in several flavors:
  - **Bounded Leakage**: The adversary may obtain leakage that in some way does not determine the entire secret key, either because the leakage is *shrinking* (as we saw in the previous class), *noisy* [NS09] (in that the key still has min-entropy conditioned on the leakage), or is *hard to invert* [DKL09] (in that the key may be statistically determined but has computational entropy conditioned on the leakage).
  - **Continual Leakage**: In which the total leakage may exceed the length of the secret key, as mentioned above.

2 Digital Signatures in the Bounded Memory Leakage Model

We start by defining a leakage-resilient signature scheme in the bounded leakage model. A digital signature scheme is a triple of PPT algorithms $\mathcal{S} = (Gen, Sign, Ver)$. Syntactically:

- $Gen(1^k) = (sk, vk)$ where k is a *security parameter* and (sk, vk) is a key-pair consisting of a private *signature key* and a public *verification key*.
- $Sign_{sk}(m) = \sigma$ where m is a *message* and σ is the *signature*.
- $Ver_{vk}(m, \sigma) \in \{0, 1\}$ takes a message and a signature and chooses whether or not to accept.

We recall the standard properties of digital signature schemes:

Correctness: The verifier accepts all honest signatures whp

$$\Pr_{(sk,vk)\leftarrow_{\mathrm{R}} Gen(1^k)}[Ver_{vk}(m, Sign_{sk}(m)) = 1] = 1 - \mathbf{negl}$$

Security under Chosen-Message Attack: We define the following forgery game between a challenger $\mathcal{C}$ and attacker $\mathcal{A}$

1. $\mathcal{C} \to \mathcal{A}$: Generates $(sk, vk) \leftarrow_{\mathrm{R}} Gen(1^k)$ and sends vk to $\mathcal{A}$.
2. $\mathcal{A} \leftrightarrow \mathcal{C}$: (Adaptively) chooses messages m_i, receives signatures $\sigma_i = Sign_{sk}(m_i)$.
3. $\mathcal{A} \to \mathcal{C}$: Outputs a message-signature pair (m^*, σ^*).

We say that $\mathcal{A}$ "wins" this game if 1) $Ver_{vk}(m^*, \sigma^*) = 1$ and 2) $m^* \neq m_i$ for every i. We say that $\mathcal{S}$ is *secure under adapative chosen-message attack* if for every PPT $\mathcal{A}$

$$\Pr_{(sk,vk)\leftarrow_{\mathrm{R}} Gen(1^k)}[\mathcal{A} \text{ wins}] = \mathbf{negl}$$

In order to model leakage we make two simple modification to the forgery game. First we have the attacker specify a (relatively short) leakage function, and then we have the challenger send back the evaluation of the leakage function on the key-pair along with the public key. Specifically we add a step 0 and modify step 1:

0. $\mathcal{A} \to \mathcal{C}$: Chooses a function $L : \{0,1\}^{|vk|+|sk|} \to \{0,1\}^{\lambda}$ and sends it to $\mathcal{C}$.

1'. $\mathcal{C} \to \mathcal{A}$: Generates $(sk, vk) \leftarrow_{\mathrm{R}} Gen(1^k)$ and sends vk and $L(vk, sk)$ to $\mathcal{A}$.

Steps 2 and 3 are the same. The criteria for $\mathcal{A}$ to win the game is the same and, similarly, we say that $\mathcal{S}$ is *secure under adaptive chosen-message attack with λ bits of leakage (henceforth, "secure")* if every PPT attacker $\mathcal{A}$ wins this game with negligible probability.

Remark 1 *More generally, we can allow the adversary to make several leakage queries, and these leakage queries can be chosen adaptively and arbitrarily interleaved with the queries to the signing oracle (step 2 of the chosen-message attack). Specifically, at any point during the chosen-message attack the adversary may request $L_i(vk, sk) \in \{0,1\}^{\lambda_i}$ subject to the constraint that $\sum \lambda_i \leq \lambda$. Although the definition we give is more restrictive, all the results presented will hold if we allow the adversary several adaptive and arbitrarily interleaved leakage queries.*

3 Overview of the Construction

In the next section we will begin describing the construction of leakage-resilient digital signatures in [KV09]. However, to give some intuition for the construction, we will begin with some intuition for why it is "impossible" to construct such a signature scheme. Suppose we want to prove that forging digital signatures in our scheme is as hard as finding collisions in an arbitrary collision-resistant hash family. The usual way to do this is assume there is a forger $\mathcal{F}$ and construct an algorithm $\mathcal{A}^{\mathcal{F}}$ that can break collision resistance. More precisely

1. $\mathcal{A}$ receives a hash function $h \in \mathcal{H}$.
2. $\mathcal{A}$ plays the forgery game with $\mathcal{F}$: $\mathcal{F}$ sends a leakage function L and $\mathcal{A}$ must send vk and some leakage value (supposedly $L(vk, sk)$) to $\mathcal{F}$.
3. Finally, $\mathcal{F}$ generates (m^*, σ^*).
4. $\mathcal{A}$ uses (m^*, σ^*) to find a collision in h.

Intuitively, since L may be given to $\mathcal{A}$ in an "obfuscated" manner, it seems that $\mathcal{A}$ can only use L as a black-box, and thus must know (vk, sk) to answer the leakage query in a way that ensures $\mathcal{F}$ will still be able to forge signatures. But if all we know is that (m^*, σ^*) given by $\mathcal{F}$ is a valid message-signature pair under (vk, sk) that are known to $\mathcal{A}$, then how can it help $\mathcal{A}$ find collisions? Indeed, if $\mathcal{A}$ knows sk then it can generate (m^*, σ^*) itself! So the key idea is to construct the scheme so as to make $\mathcal{F}$ produce (m^*, σ^*) that is valid under a *different* signing key sk', and that (sk, sk') will be a collision in the hash function h.

The construction will make use of the following cryptographic primitives

- A standard semantically-secure encryption scheme $\mathcal{E} = (Gen_{\mathcal{E}}, Enc, Dec)$.
- A shrinking collision-resistant hash family $\mathcal{H} = \{\mathcal{H}_k\}$ where $\mathcal{H}_k \ni h : \{0,1\}^{k^t} \to \{0,1\}^k$ (for a constant $t > 1$ to be chosen later).
- A simulation-sound $\mathcal{NIZK}$ proof system $(\ell, \mathcal{P}, \mathcal{V}, \mathcal{S}_1, \mathcal{S}_2)$ for $\mathcal{NP}$.

The first two primitives are standard, but we elaborate on the final one in the next section.

3.1 Non-Interactive Zero Knowledge Proofs

A non-interactive zero-knowledge proof system has all the standard properties of zero-knowledge proofs except that the prover only sends one message to the verifier. Such proof systems are always constructed in the *common random string* model, where both the prover and verifier are given access to a shared string $\text{CRS} \in \{0,1\}^{\ell(k)}$. A $\mathcal{NIZK}$ proof system for a language L consists of $(\ell, \mathcal{P}, \mathcal{V}, \mathcal{S}_1, \mathcal{S}_2)$ where ℓ is a polynomial and $\mathcal{P}, \mathcal{V}, \mathcal{S}_1, \mathcal{S}_2$ are PPT algorithms. We require the following properties:

Completeness: All true statements can be proven. Let R_L denote the witness relation for L, that is $(x, w) \in R_L$ if $x \in L$ and w is an $\mathcal{NP}$-witness for x. Then completeness states that $\forall k$, $\forall x \in L$ s.t. $|x| = \text{poly}(k)$, $\forall w$ s.t. $(x, w) \in R_L$, and $\forall \text{CRS} \in \{0,1\}^{\ell(k)}$

$$\mathcal{V}(x, \mathcal{P}(x, w, \text{CRS}), \text{CRS}) = 1.$$

(Standard) Soundness: Only true statements can be proven. For every $\mathcal{A}$

$$\Pr_{\substack{\text{CRS} \leftarrow_R \{0,1\}^{\ell(k)} \\ \text{coins}}} [\mathcal{A}(\text{CRS}) = (x, \pi) \text{ s.t. } x \notin L \text{ and } \mathcal{V}(x, \pi, \text{CRS}) = 1] = \textbf{negl}$$

Zero-Knowledge: Let $\text{REAL}_{\mathcal{A}}$ denote the distribution on transcripts of $\mathcal{A}^{\mathcal{P}(\cdot,\cdot,\text{CRS})}(\text{CRS})$. That is, the interaction of $\mathcal{A}$ with the honest-prover, wrt to a given reference string. We want to claim that $\text{REAL}_{\mathcal{A}}$ is computationally indistinguishable from the interaction of $\mathcal{A}$ with the simulators. Let $\mathcal{S}'(x, w, \text{CRS}, \tau) = \mathcal{S}_2(x, \text{CRS}, \tau)$ whenever $(x, w) \in R_L$, and $\mathcal{S}' = \perp$ otherwise.[1] Let $\text{IDEAL}_{\mathcal{A}}^{\mathcal{S}_1,\mathcal{S}_2}$ be the following distribution on transcripts: Use $\mathcal{S}_1$ to choose a reference string and a *trapdoor* $(\text{CRS}, \tau) \leftarrow_{\text{R}} \mathcal{S}_1(1^k)$. Run $\mathcal{A}^{\mathcal{S}'(\cdot,\cdot,\text{CRS},\tau)}(\text{CRS})$. We say the proof system is zero-knowledge if

$$\text{REAL}_{\mathcal{A}} \approx_C \text{IDEAL}_{\mathcal{A}}^{\mathcal{S}_1,\mathcal{S}_2}$$

where $\approx_C$ denote computational indistinguishability.

Simulation-Soundness: For our scheme we require one additional security property called *simulation-soundness*. Informally, we don't want an adversary $\mathcal{A}$ to be able to violate the soundness of the proof system *even after interacting with the simulator* (who knows the trapdoor). We formalize this using the following game

1. $(\text{CRS}, \tau) \leftarrow_{\text{R}} \mathcal{S}_1(1^k)$.
2. $\mathcal{A}^{\mathcal{S}_2(\cdot,\text{CRS},\tau)}(\text{CRS})$ makes queries x_i to $\mathcal{S}_2$.
3. $\mathcal{A}^{\mathcal{S}_2(\cdot,\text{CRS},\tau)}(\text{CRS}) \to (x^*, \pi^*)$.

We say $\mathcal{A}$ wins this game if 1) $x^* \neq x_i$ for every i, 2) $x^* \notin L$, and 3) $\mathcal{V}(x^*, \pi^*, \text{CRS}) = 1$. As expected, we say the proof system is simulation-sound if for every $\mathcal{A}$

$$\Pr_{(\text{CRS},\tau)\leftarrow_{\text{R}}\mathcal{S}_1(1^k)}[\mathcal{A} \text{ wins}] = \mathbf{negl}$$

4 Construction of Leakage-Resilient Signatures

We will now describe Katz and Vaikuntanathan's construction of leakage-resilient digital signatures. Recall we let $\mathcal{E} = (Gen_{\mathcal{E}}, Enc, Dec)$ be a semantically-secure encryption scheme and $\mathcal{H} = \{\mathcal{H}_k\}$ be a shrinking collision-resistant hash family from $\{0,1\}^{k^t} \to \{0,1\}^k$. We also let $(\ell, \mathcal{P}, \mathcal{V}, \mathcal{S}_1, \mathcal{S}_2)$ be a simulation-sound $\mathcal{NIZK}$ proof-system for $\mathcal{NP}$. Specifically the scheme needs to prove membership in the language

$$L = \{(m, h, y, pk, c) \mid \exists (x, r)\ c = Enc_{pk}(x; r), h(x) = y\}.$$

Less formally, the signatures will include a proof that c is an encryption of a preimage of y under the function $h \in \mathcal{H}$. Notice that the message m is not used anywhere in specifying the language! However, we will use the simulation-soundness of the $\mathcal{NIZK}$ to argue that its infeasible to produce proofs for new statements $(m', h, y, pk, c) \in L$ even when all the other inputs are the same!

We can now specify the scheme:

[1] The reason we don't simply use $\mathcal{S}_2$ in place of $\mathcal{S}'$ is because we need $\mathcal{S}'$ only to work for valid witnesses, as the honest prover would.

Gen(1^k)**:**
$h \leftarrow_{\mathrm{R}} \mathcal{H}_k$, $x \leftarrow_{\mathrm{R}} \{0,1\}^{k^t}$, CRS $\leftarrow_{\mathrm{R}} \{0,1\}^{\ell}$, $pk \leftarrow_{\mathrm{R}} Gen_{\mathcal{E}}(1^k)$[2]
$y \leftarrow h(x)$
Output the key pair (vk, sk) where $sk = x$, $vk = (h, y, pk, \mathrm{CRS})$

Sign$_{sk}(m)$**:**
$r \leftarrow_{\mathrm{R}} \{0,1\}^*$, $c \leftarrow Enc_{pk}(x; r)$
$\pi \leftarrow \mathcal{P}((m, h, y, pk, c), (x, r), \mathrm{CRS})$
Output the signature $\sigma = (c, \pi)$

Ver$_{vk}(m, \sigma = (c, \pi))$**:**
Output $\mathcal{V}((m, h, y, pk, c), \pi, \mathrm{CRS})$

Theorem 2 *The scheme above is secure under adaptive chosen-message attack with λ bits of leakage when $\lambda \leq k^t - k = (1 - k^{1-t})|sk|$.*

Proof:

The security proof works by reduction to finding collisions in the hash family $\{\mathcal{H}_k\}$. Let $\mathcal{F}$ be a signature forger for this scheme. We construct an adversary $\mathcal{A}$ that takes as input $h \in \mathcal{H}_k$ and uses $\mathcal{F}$ to find collisions in h (with non-negligible probability). To this end, $\mathcal{A}$ plays the role of the challenger in the forgery game with $\mathcal{F}$ as follows: $\mathcal{A}$ computes a signature key $sk = x \leftarrow_{\mathrm{R}} \{0,1\}^{k^t}$ and $y = h(x)$, a key pair $(pk_{\mathcal{E}}, sk_{\mathcal{E}}) \leftarrow_{\mathrm{R}} Gen_{\mathcal{E}}(1^k)$ for the encryption scheme, and a reference string CRS $\leftarrow_{\mathrm{R}} \{0,1\}^{\ell}$. $\mathcal{A}$ sends $vk = (h, y, pk_{\mathcal{E}}, CRS)$ and a leakage $L(vk, sk)$ to $\mathcal{F}$ and answers $\mathcal{F}$'s queries to the signing oracle. If $\mathcal{F}$ breaks the security of the signature-scheme, then it produces a signature pair (m^*, σ^*) where σ^* contains an encryption of a preimage x' such that $h(x') = y$. We prove that whp $x \neq x'$ and thus $\mathcal{A}$ can use $sk_{\mathcal{E}}$ to decrypt x' and find a collision in h. This intuition can be formalized in two steps:

Step 1: Suppose that $\mathcal{A}$ and $\mathcal{F}$ play the forgery game but $\mathcal{F}$ makes no chosen-message queries. That is, $\mathcal{F}$ sends L to $\mathcal{A}$, $\mathcal{A}$ generates a key-pair (vk, sk) for the signature scheme and sends $vk, L(vk, sk)$ to $\mathcal{F}$ and then $\mathcal{F}$ returns (m^*, σ^*) such that $Ver_{vk}(m^*, \sigma^*) = 1$.

Observe that, from the soundness of the $\mathcal{NIZK}$ proof, $\sigma^* = (c, \pi)$ such that $c = Enc_{pk}(x')$ where $h(x') = y = h(x)$. Now if we can show that $x' \neq x$ whp then we'll be done (at least in this scenario). This fact follows from an information theoretic argument. Recall that x is a k^t-bit string, and the only information that $\mathcal{F}$ has about x is $L(vk, sk) = L(vk, x)$, which is λ bits long, and $y = h(x)$, which is k bits long. Now we invoke the following lemma that ensures x is still highly uncertain to $\mathcal{F}$ given the $k + \lambda$ bits of information $\mathcal{F}$ has seen.

Lemma 3 *Let X be a random variable with $H = \mathrm{H}_\infty(X)$ and f be a function with ℓ-bit range. Then*

$$\Pr_X [\mathrm{H}_\infty(X|f(X)) \leq 1] \leq 2^{\ell - H - 1}.$$

Applying this lemma where f is the leakage and the image $y = h(x)$, $H = k^t$, $\ell = k + \lambda \leq k^t$ tells us that with probability at least $1/2$, x has at least 1 bit of min-entropy leftover in $\mathcal{F}$'s view. If x has at least 1 bit of min-entropy then the probability that $\mathcal{F}$ produces $x' = x$ is at most $1/2$. So with

[2]Here we assume it is possible to sample a public-key pk from $Gen_{\mathcal{E}}$ with the correct marginal distribution.

probability at least 1/4 times the success probability of $\mathcal{F}$, $\mathcal{A}$ can produce a collision in h, violating collision resistance when $\mathcal{F}$ has a non-negligible probability of success.

Step 2: Notice that, at least information theoretically, sending an encryption $c = Enc_{pk}(x)$ determines x. Thus if we do allow $\mathcal{F}$ to do a chosen-message attack, from his view x will be determined and we won't be able to argue as in Step 1. We can fix this problem with two changes to the way $\mathcal{A}$ responds to $\mathcal{F}$'s chosen-message queries:

- Instead of computing $c = Enc_{pk}(x)$, compute $c = Enc_{pk}(0)$.
- Instead of computing $\pi = \mathcal{P}((m_i, h, y, pk, c), (x, r), \text{CRS})$, have $\mathcal{A}$ generate $(\text{CRS}, \tau) \leftarrow_{\text{R}} \mathcal{S}_1(1^k)$ and compute $\pi = \mathcal{S}_2((m_i, h, y, pk, c), \text{CRS}, \tau)$.

Note that the first change has only a negligible affect on $\mathcal{F}$'s ability to forge a signature because *Enc* is semantically-secure, and the second change has only a negligible affect on $\mathcal{F}$'s ability to forge a signature because the zero-knowledge property of the $\mathcal{NIZK}$ says that the simulated proofs are indistinguishable from the actual proofs.

More specifically, at the end of the chosen-message attack, $\mathcal{F}$ outputs a pair $(m^*, \sigma^*) = (m^*, c^*, \pi^*)$ and with non-negligible probability, this is a valid message-signature pair. Then π^* is an accepting proof for the statement $(m^*, h, y, pk, c^*) \in L$. But the conditions of a chosen-message attack say that $\mathcal{F}$ must never have asked for a signature of m^* and thus has never seen a proof of any statement for which m^* is the message. By simulation-soundness, his proof is valid, and thus $Dec_{sk}(c^*) = x' \in h^{-1}(h(x))$. Now, since $Enc_{pk}(0)$ contains no information about x, we can repeat the argument from Step 1 to show that $x \neq x'$ with constant probability.

□

5 Preview of Continual Leakage Resilience

In the next class we will see how to construct cryptographic primitives that are secure even when the adversary gets to see leakage that is larger than the secret key. More specifically, we will consider a key-update function f and a sequence of secret keys $sk_1, sk_2, sk_3, \ldots$ where $sk_i = f(sk_{i-1})$. The adversary can specify a sequence of leakage functions $L_1, L_2, L_3, \ldots$ and gets to see $L_i(sk_i)$ after each update. Here we assume a bound on the length of each leakage function L_i but not on the number of times we update the key and incur leakage. In other words, we only bound the *rate* of leakage and not the *total* leakage. To make our job even harder, we'd like the adversary to be oblivious to how many times the key has been updated. In particular, we don't want the length of encryptions or signatures to be growing with the number of updates and, more importantly, we don't want to have to change the public key every time we update.

Let's give some intuition for why this notion of security might be hard to achieve. Consider the scheme we just constructed in which the secret key is x and the public key contains $y = h(x)$ where h is from a collision-resistant hash family. Then in order to get a new secret key that works for the public key y, we have to find $x' \neq x$ such that $y = h(x) = h(x')$. But of course, that would violate collision-resistance. In fact, this update would violate exactly the property we used to prove security of the scheme! Similarly, if we recall the leakage-resilient encryption scheme of Gentry, Peikert, and Vaikuntanathan [GPV08], the secret key was $r \in \{0,1\}^k$ and the public key was (A, rA) where $A \in \mathbb{Z}_q^{m \times n}$. Then to find a new secret key for the same public key we must find $r' \neq r$ such that

$(r - r')A = 0$. But again, the inability to find such an r' was exactly what we used to prove security of the GPV scheme!

In the next class we will see how to overcome this difficulty:

- Let $A = A_{pk}$ be the set of possible secret keys corresponding to a public key pk. Then the scheme will make all its updates to the secret key occur in some neighborhood $A' = A'_{pk,sk} \subseteq A$ around the initial secret key $sk \in A$. We want it to be feasible to do the updates (find new secret keys $sk_1, sk_2, \cdots \in A'$, but infeasible to find a secret key $sk' \in A \setminus A'$.
- From the view of the adversary, the location of A' is statistically hidden, so that an adversary trying to break security of the protocol will have to produce $sk \in A \setminus A'$.

Next class we will see a sketch of how this approach can be made into a working scheme.

References

[AGV09] A. Akavia, S. Goldwasser, V. Vaikuntanathan. *Simultaneous Hardcore Bits and Cryptography Against Memory Attacks*. In TCC2009.

[DKL09] Y. Dodis, Y. Kalai, S. Lovett. *On Cryptography with Auxiliary Input*. In STOC2009.

[GPV08] C. Gentry, C. Peikert, V. Vaikuntanathan. *Trapdoors for Hard Lattices and New Cryptographic Constructions*. In STOC2008.

[KV09] J. Katz, V. Vaikuntanathan. *Signatures with Bounded Leakage Resilience*. In ASIACRYPT2009.

[MR04] S. Micali, L. Reyzin. *Physically Observable Cryptography*. In TCC2004.

[NS09] M. Naor, G. Segev. *Public-Key Cryptosystems Resilient to Key Leakage*. In CRYPTO2009.