

Revisiting TESLA in the quantum random oracle model



Erdem Alkim
Nina Bindel
Johannes Buchmann
Özgür Dagdelen
Edward Eaton
Gus Gutoski
Juliane Krämer
Filip Pawlega

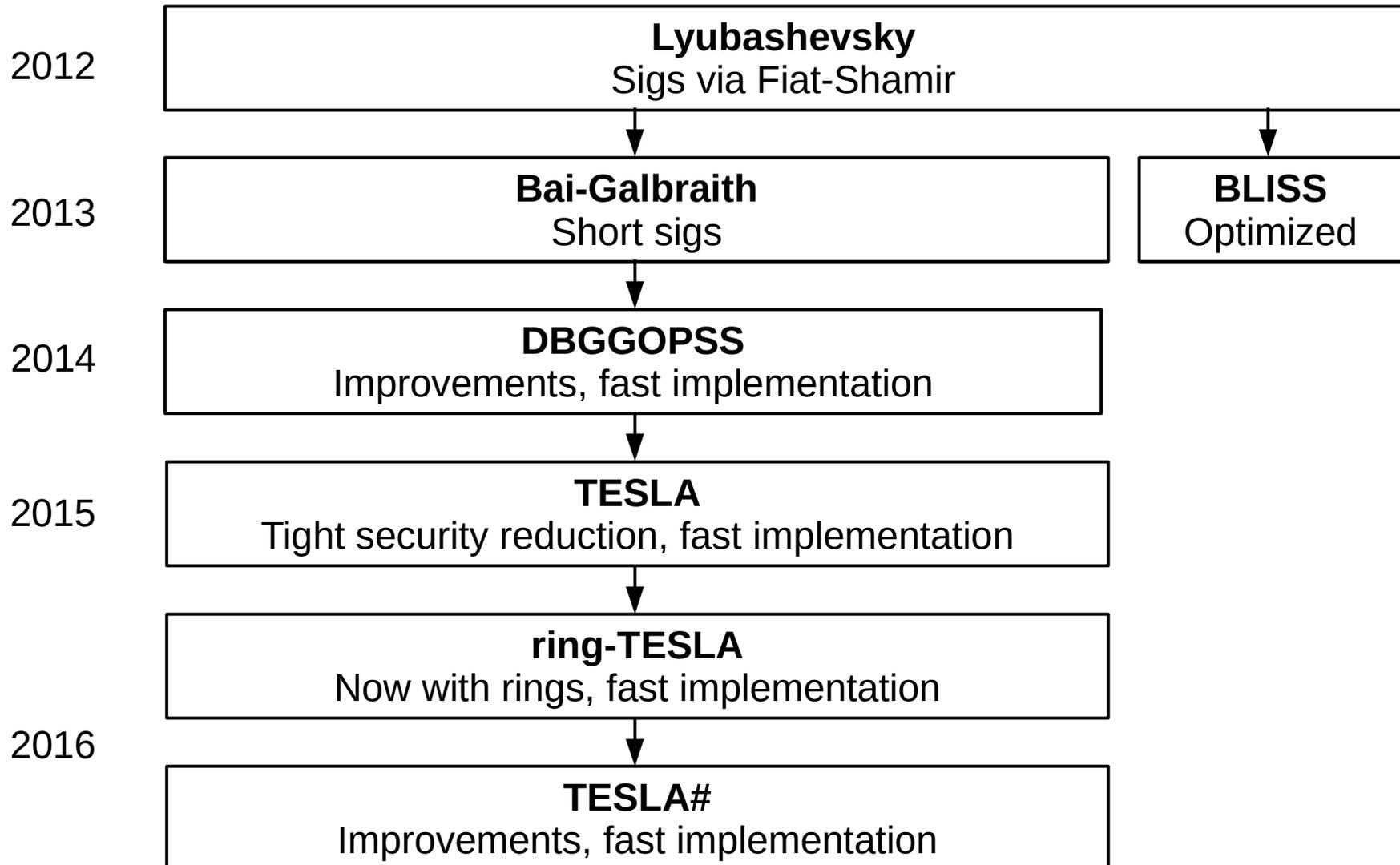


TECHNISCHE
UNIVERSITÄT
DARMSTADT

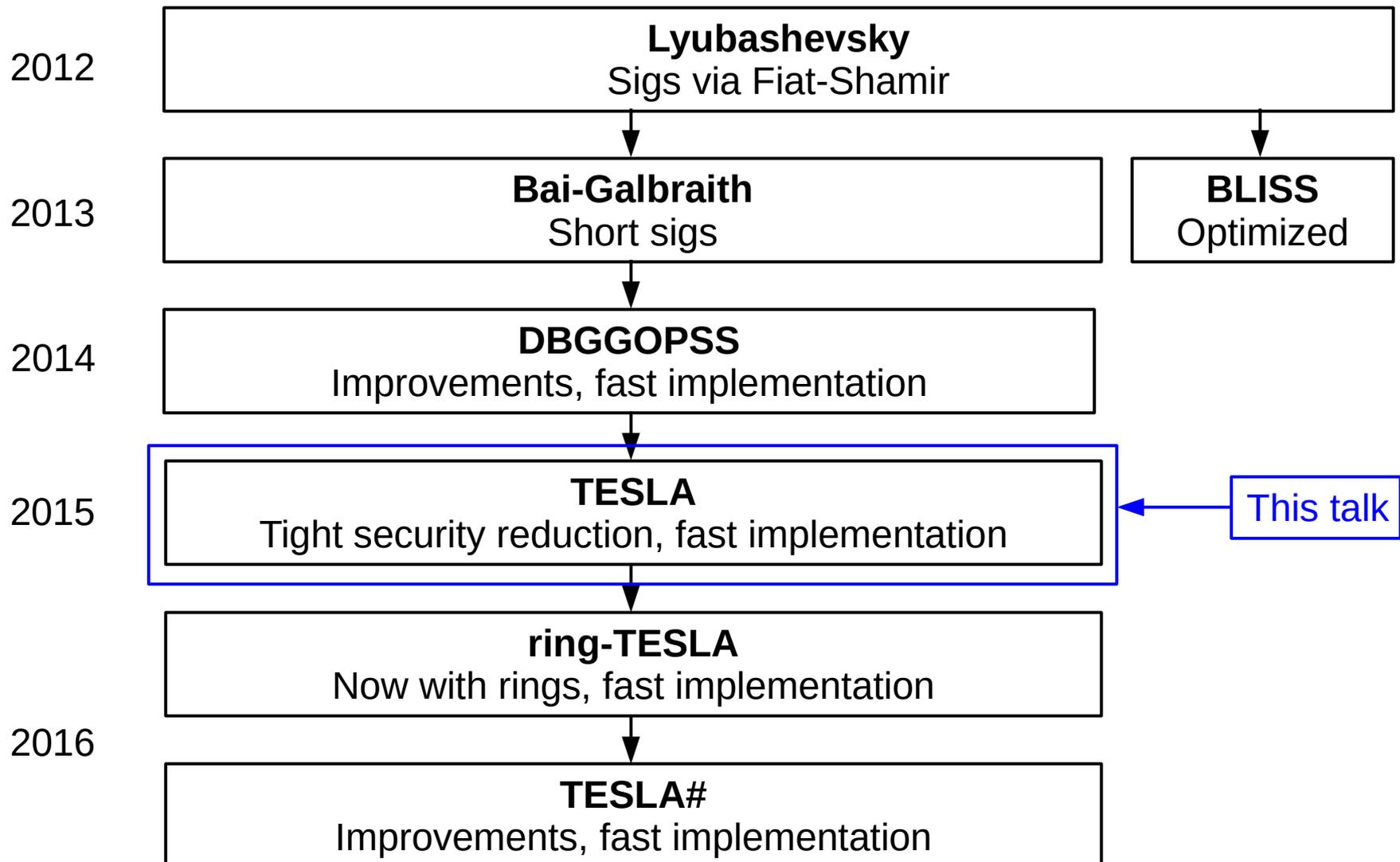


Full version available as
Cryptology ePrint Archive: Report 2015/755
<https://ia.cr/2015/755>

Selected history of Fiat-Shamir— style signatures from LWE or SIS



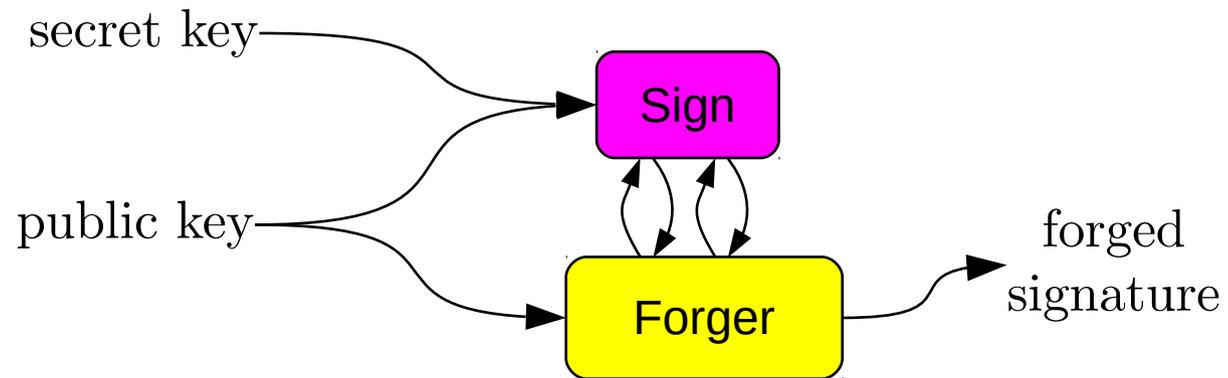
Selected history of Fiat-Shamir— style signatures from LWE or SIS



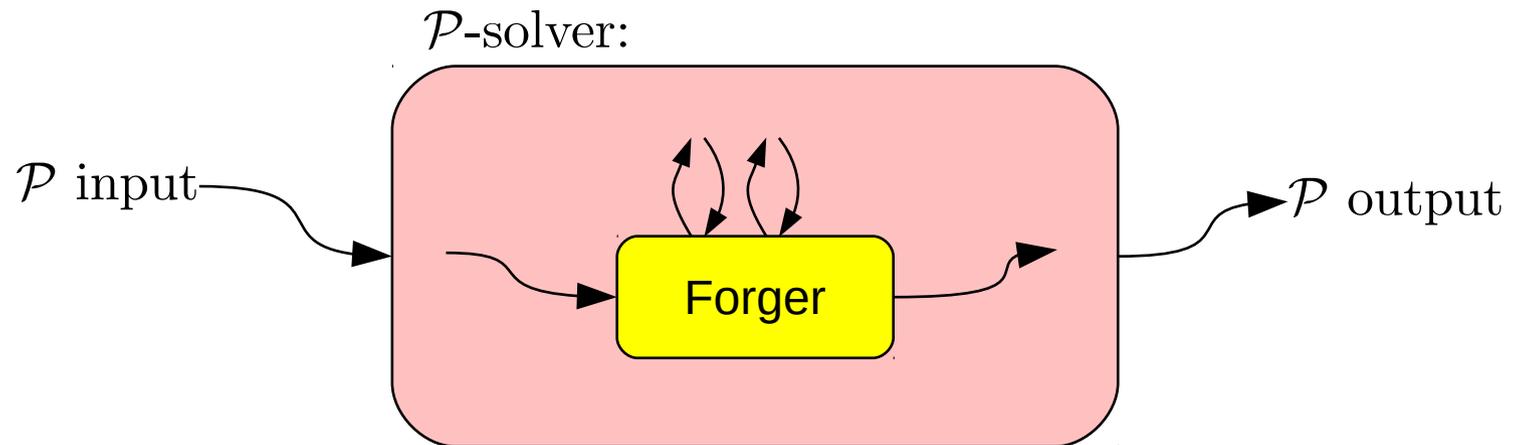
Preamble

- Parameter selection, tightness.
- The quantum random oracle model (QROM).

Given a forger...



...construct a \mathcal{P} -solver



Parameter choice should account for the security reduction

$$\begin{aligned} & (\mathcal{P}\text{-solver run time}) \\ &= (\text{forger run time}) + (\text{reduction run time}) \end{aligned}$$

Assumption: Problem \mathcal{P} cannot be solved in time less than t .

$$(\text{forger run time}) \geq t - (\text{reduction run time})$$

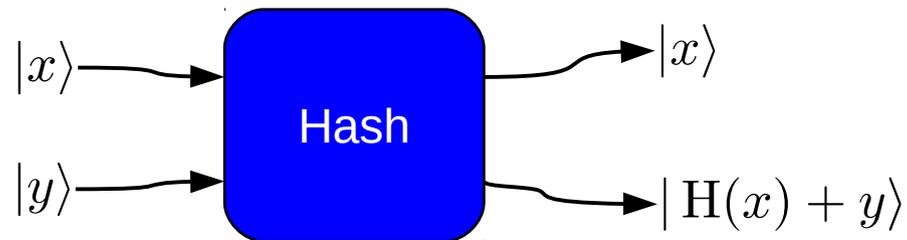
Important: Choose parameters so that $t - (\text{reduction run time})$ is intractable.

Tightness

- If (reduction run time) is small then the reduction is *tight*.
- All else equal, tight is preferred to non-tight:
 - Superior efficiency for a given level of security.

The quantum random oracle model (QROM)

- A quantum forger can query the random oracle in *quantum superposition*.

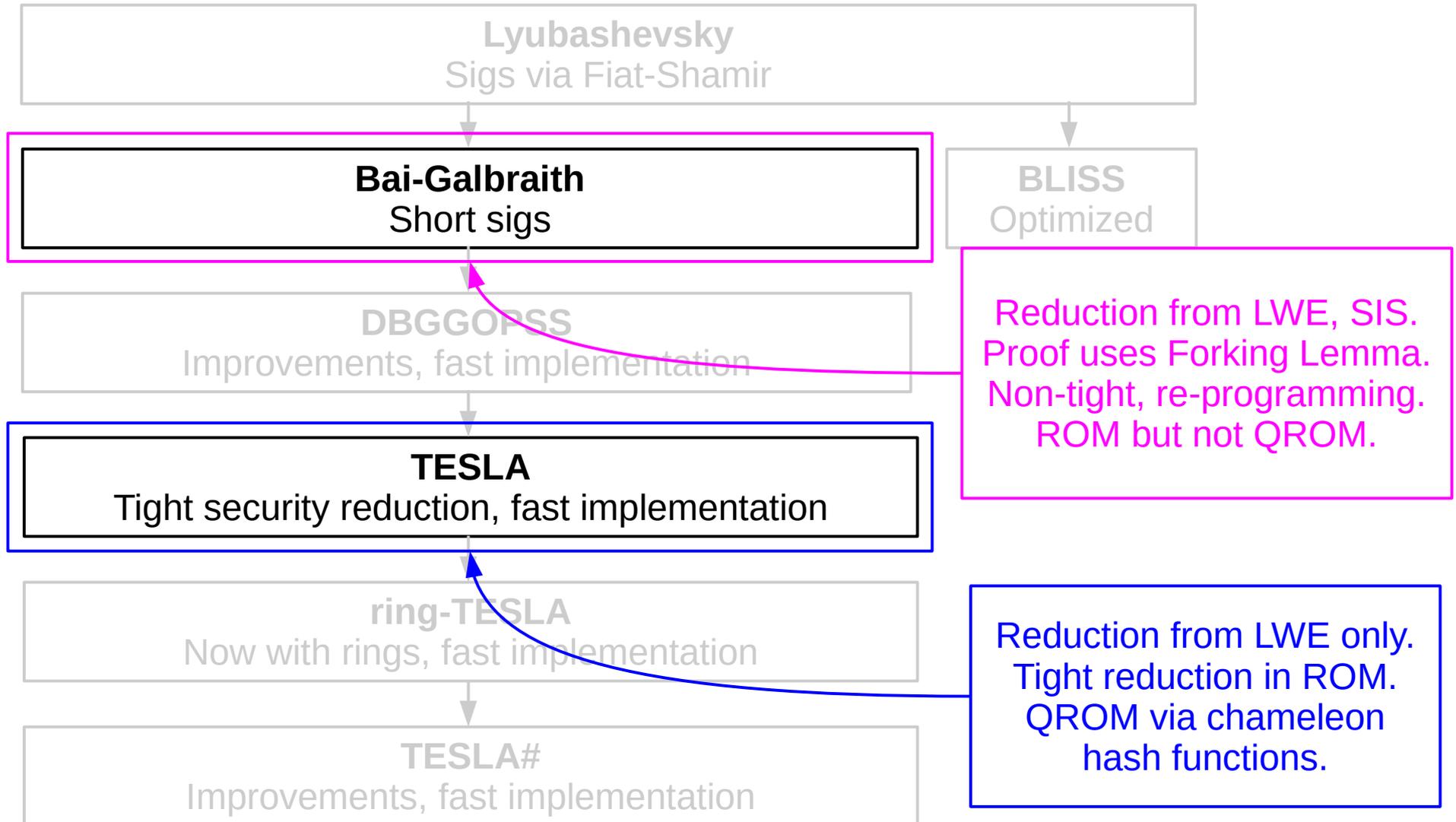


- It is conceivable that a scheme is secure in ROM but not in QROM.
- For a scheme to be quantum-resistant, its security reduction must hold in the QROM.

When does ROM imply QROM?

- [BDFLSZ-2011]: ROM \implies QROM if the reduction is *history-free*.
- Many ROM proofs involve *re-programming* the random oracle.
 - Not history-free.
- There is little research on QROM + re-programming. [Unruh-2014, ES-2015, Unruh-2017]

Prior work on TESLA



Our contributions (theoretical)

- The 2015 TESLA security proof is flawed. (Also noticed by Chris Peikert.)
- New, tight security reduction from LWE.
 - Completely re-done from scratch.
- Direct reduction in the QRROM with re-programming.
 - No need for chameleon hash functions.
- Bonus: Proofs of Gaussian heuristic for q -ary lattices.

Our contributions (practical)

- New parameter sets chosen according to our tight security reduction.

| | |
|---------|--------------------|
| TESLA-0 | 96 bits classical |
| TESLA-1 | 128 bits classical |
| TESLA-2 | 128 bits quantum |

- Software implementation of TESLA-0, TESLA-1 targeting Intel Haswell CPU.

Summary of related work

- Proof approach: [KW-2003], [AFLT-2012].
- Other tight LWE/SIS sigs: [GPV-2008], [BL-2016] (trapdoor), [AFLT-2012].

Katz, Wang

Abdalla, Fouque, Lyubashevsky, Tibouchi

Gentry, Peikert, Vaikuntanathan

Boyer, Li

“Lattice-based” crypto

- **Fact:** $\text{LWE/SIS} \geq \text{GapSVP, SIVP}$.
- These are *worst-case to average-case reductions* from fundamental hard problems on lattices.
- However, these reductions are non-tight.
- Parameter sets are never selected according to these reductions.
- Our TESLA parameter sets are based on hardness of LWE, not GapSVP or SIVP.

“Lattice-based” crypto

TESLA:

Tightly-secure, Efficient signature scheme from Standard Lattices.

TESLA:

Tightly-secure, Efficient Signature scheme from Learning with Errors.

Learning with Errors (LWE) (matrix version)

Input. Matrices $A \in \mathbb{Z}_q^{m \times n}$ and $T \in \mathbb{Z}_q^{m \times n'}$.

- Entries of A are uniformly random.

Yes. $T = AS + E$

- Entries of $S \in \mathbb{Z}_q^{n \times n'}$ and $E \in \mathbb{Z}_q^{m \times n'}$ sampled from a discrete Gaussian on \mathbb{Z}_q .
- (S, E) is a *witness*.

No. Entries of T are uniformly random.

TESLA key generation

Pk: LWE yes-instance Sk: witness

1. Choose LWE witness (S, E) with Gaussian entries.
2. Check: If entries of S, E are too large then restart.
3. Choose A uniformly at random.
4. $T \leftarrow AS + E$
5. Return pk: (A, T) , sk: (S, E) .

TESLA sign

Zero-knowledge proof (S,E) + Fiat-Shamir

Input. sk: (S, E) , pk: (A, T) , msg.

1. Choose a random “short” vector $y \in \mathbb{Z}_q^n$.
2. $c \leftarrow \text{H}(\text{hi-bits}(Ay), \text{msg})$.
3. If $Ay - Ec$ is not “well-rounded” then restart.
4. $z \leftarrow y + Sc$.
5. If z is not “short” then restart.
6. Return signature (z, c) .

TESLA sign: terminology

- Range of $H(\cdot)$ is vectors in $\{-1, 0, 1\}^{n'}$ of bounded weight.
 - Entries of Ec, Sc are guaranteed to be bounded.
- w is *well-rounded* if entries of w , $\text{lo-bits}(w)$ are not too large.
 - Signatures always verify.
 - Signatures do not leak info on the secret key.

TESLA verify

Input. pk: (A, T) , sig: (z, c) , msg.

1. If z is not “short” then reject.

2. Accept $\iff c = H(\text{hi-bits}(Az - Tc), \text{msg})$.

Observe.

$$Az - Tc = Ay - Ec$$

$$\text{hi-bits}(Ay - Ec) = \text{hi-bits}(Ay)$$

due to well-roundedness.

Security theorem for TESLA

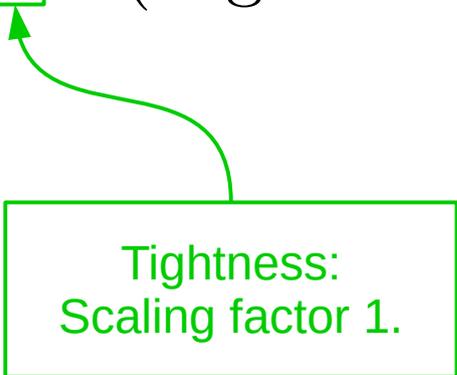
Theorem. Matrix-LWE is (t, ε) -hard \implies TESLA is (t', ε') -unforgeable in QRROM with $t' \lesssim t$ and

$$\varepsilon' \leq \varepsilon + (\text{negl in TESLA params})$$

Security theorem for TESLA

Theorem. Matrix-LWE is (t, ε) -hard \implies TESLA is (t', ε') -unforgeable in QRROM with $t' \lesssim t$ and

$$\varepsilon' \leq \boxed{\varepsilon} + (\text{negl in TESLA params})$$

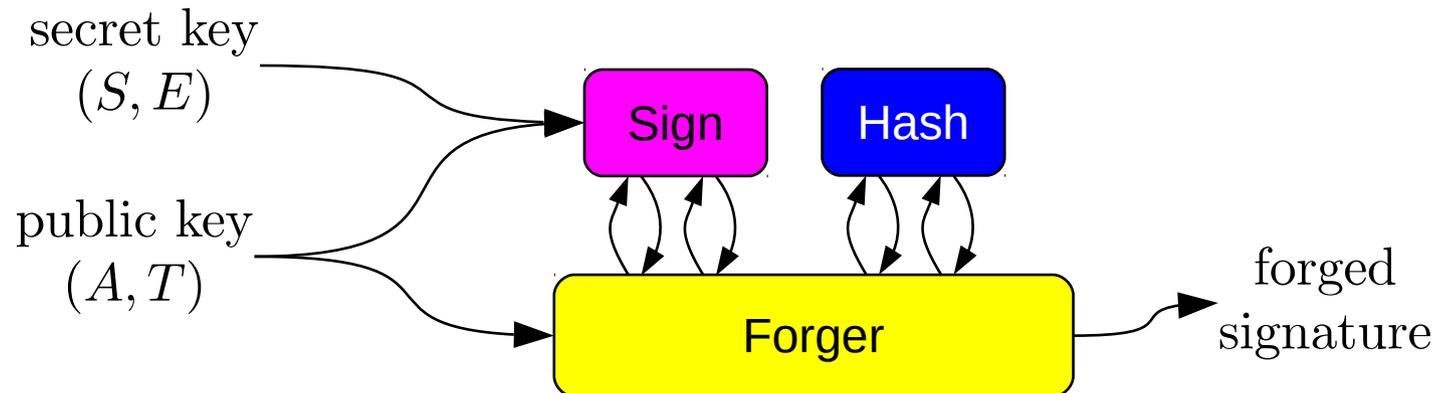


Tightness:
Scaling factor 1.

Proof overview

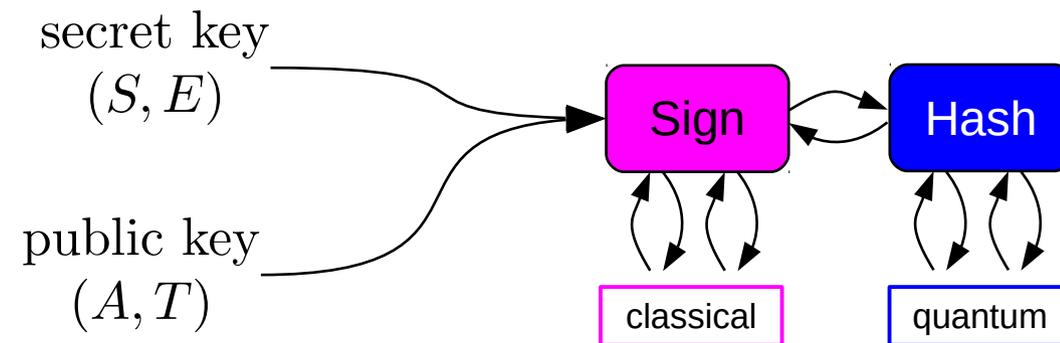
Follow the lead of [KW-2003], [AFLT-2012]; make it work in QRROM.

Suppose there is a TESLA forger:

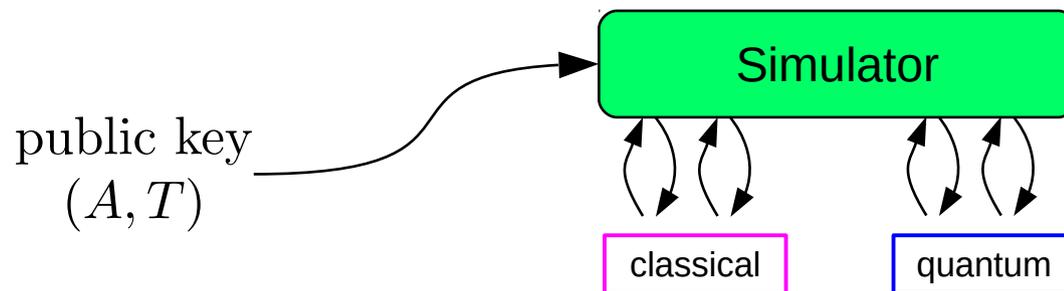


Simulator

Build a simulator for TESLA signatures.



Real sign,
hash oracles.

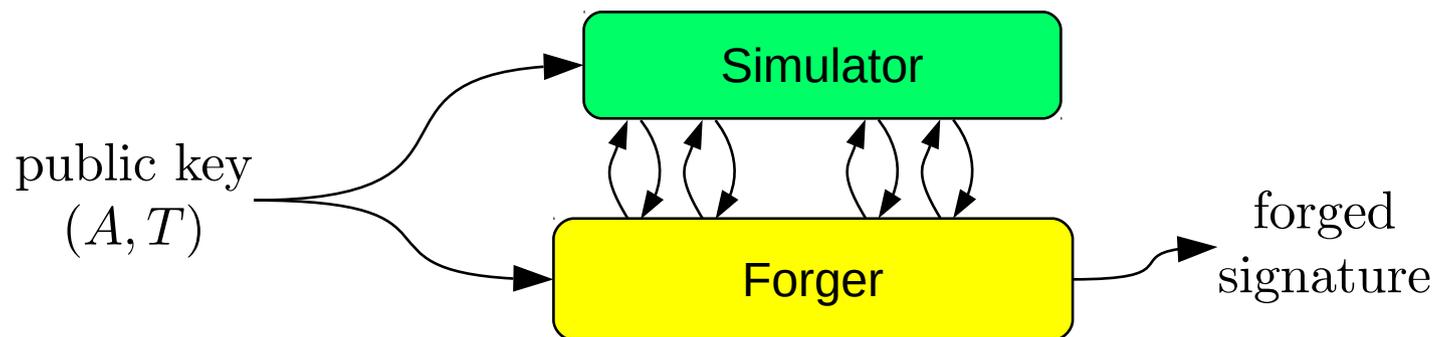


Simulated oracles.

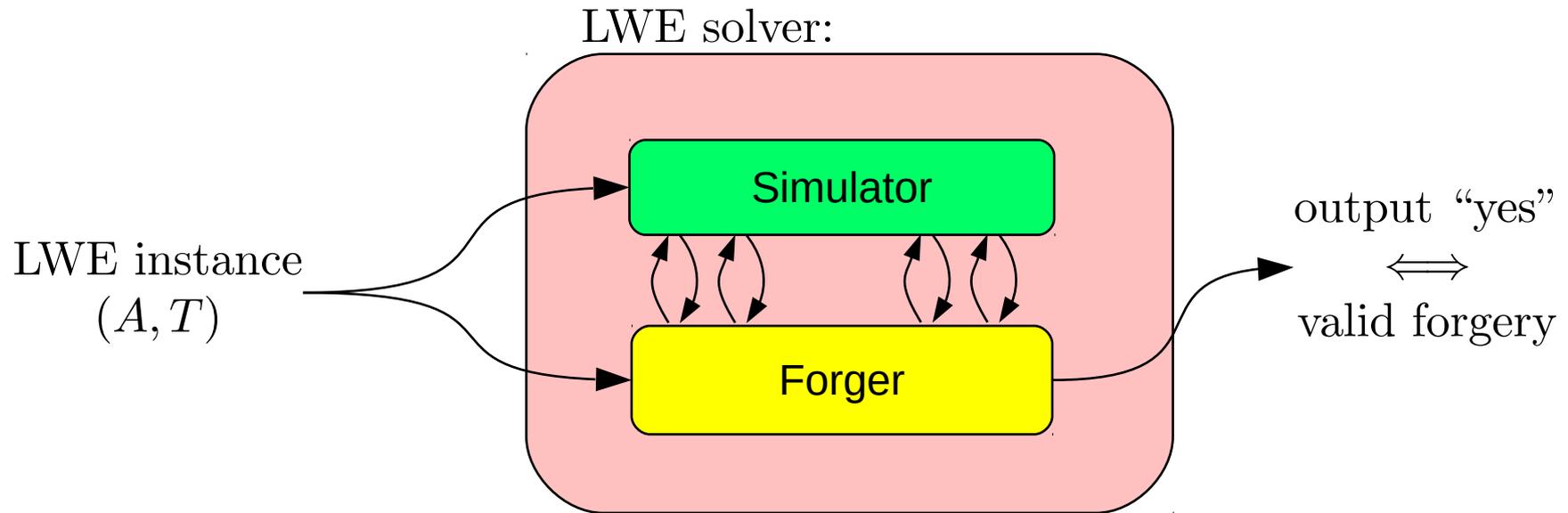
Forger forges, even with a simulator

If simulation is accurate then

$$\begin{aligned} & \text{output of (forger + real)} \\ & \approx \text{output of (forger + sim)} \end{aligned}$$



Forger + Simulator = LWE solver



Forger + Simulator = LWE solver

- If (A, T) is a LWE yes-instance:

$$\text{forger} + \text{simulator} = \text{forgery}$$

\implies output “yes”.

- **Need to prove:** If (A, T) is a LWE no-instance:

$$\text{forger} + \text{simulator} \neq \text{forgery}$$

\implies output “no”.

Yes-instances: Signature simulator

Input. $\text{pk}: (A, T), \text{msg}.$

1. $z \leftarrow$ random “short” element of \mathbb{Z}_q^n .
2. $c \leftarrow$ random $c \in \{-1, 0, 1\}^{n'}$ of bounded weight.
3. If $Az - Tc$ is not “well-rounded” then restart.
4. Re-program $H(\text{hi-bits}(Az - Tc), \text{msg}) \leftarrow c$.
5. Return signature (z, c) .

Yes-instances: Signature simulator

Input. $pk: (A, T), \text{msg}.$

1. $z \leftarrow$ random “short” element of \mathbb{Z}_q^n .
2. $c \leftarrow$ random $c \in \{-1, 0, 1\}^{n'}$ of bounded weight.
3. If $Az - Tc$ is not “well-rounded” then restart.
4. **Re-program** $H(\text{hi-bits}(Az - Tc), \text{msg}) \leftarrow c$.
5. Return signature (z, c) .

Re-program a quantum oracle!

Re-programming in TESLA

ρ_H : State prepared with t queries to $H(\cdot)$.

$H'(\cdot)$: Agrees with $H(\cdot)$ except on a small number of inputs (\cdot, msg) for each msg.

Theorem. $\|\rho_{H'} - \rho_H\|_1 < \gamma$ except w/Pr

$$\frac{t^2}{\gamma^2} \times (\text{negl in TESLA params}).$$

Proof. [BBBV-1996] + Markov's inequality + gymnastics. □

No-instances: Good hash inputs

- Ability to forge \implies can find (w, msg) whose hash $c = H(w, \text{msg})$ satisfies:

$$\exists \text{ short } z \text{ with } \text{hi-bits}(Az - Tc) = w. \quad (1)$$

- $\forall (w, \text{msg})$: The hash of (w, msg) obeys (1) with prob

$$\frac{\#\{c \text{ with } (1)\}}{\#\{\text{all } c\}}$$

over the choice of random oracle $H(\cdot)$.

Search through unstructured space

- Need to prove: $\#\{c \text{ with } (1)\}$ is small for LWE no-instances.
- Then: each (w, msg) leads to a forgery with negligible probability, independent of all others.
- The only way to find such a (w, msg) is by search through unstructured space.
- Apply lower bounds for quantum search. (Grover is optimal.)
- Forging for LWE no-instances requires many hash queries.

Good hash inputs are rare

Theorem. If TESLA params are “convenient” then

$$\text{Ex}_{(A,T)} \left[\max_w \#\{c \text{ with } (1)\} \right] \leq 1.$$

Proof. Take differences, swap summations. □



Parameter sets

| | |
|---------|--------------------|
| TESLA-0 | 96 bits classical |
| TESLA-1 | 128 bits classical |
| TESLA-2 | 128 bits quantum |

- Warning: parameters are large, TESLA is not yet ready for practical use.
- Our priority is to establish a correct security reduction in QRROM.
- That said, TESLA is far more efficient than all other schemes whose parameter choice accounts for a reduction from LWE/SIS.

Parameter sets

| | TESLA-0 | TESLA-1 | TESLA-2 |
|------|---------------|---------------|------------------|
| n | 644 | 804 | 1300 |
| n' | 390 | 600 | 1036 |
| m | 3156 | 4972 | 4788 |
| q | $2^{31} - 99$ | $2^{31} - 19$ | $\approx 2^{36}$ |
| pk | 4.6 MB | 11.2 MB | 21.8 MB |
| sk | 1.8 MB | 4.2 MB | 7.7 MB |
| sig | 1.8 KB | 2.3 KB | 4.0 KB |

(Key sizes would be much smaller in ring-TESLA.)

$$\begin{aligned} A &\in \mathbb{Z}_q^{m \times n} & S &\in \mathbb{Z}_q^{n \times n'} \\ E, T &\in \mathbb{Z}_q^{m \times n'} & z &\in \mathbb{Z}_q^n \end{aligned}$$

Software

- Targets the Intel Haswell microarchitecture.
- Based on software from [DBGGOPSS-2014].
- Use vectorized instructions where possible.
- TESLA-2 params are too big for intermediate computations to fit into a 53-bit mantissa
 - Cannot use the same code as TESLA- $\{0,1\}$.

<https://tesla.informatik.tu-darmstadt.de/de/tesla/>

Thank you!

Global A matrix?

- **Alternative:** Make A a fixed, global parameter.
- **Pros:** Smaller public keys, no expensive seed expansion.
- **Cons:** Potential back door, all-for-the-price-of-one attacks.

Proof approach

- [KW-2003]: Tightly secure signatures from DDH.
- [AFLT-2012]: Transform “lossy” ID schemes into tightly secure signatures.
 - ROM proof involves re-programming. Not history-free. Not known to hold in QROM.
 - TESLA could fit into this framework.
 - Need to re-prove AFLT in the QROM.

Other tightly-secure LWE or SIS signatures (move to the end?)

- [GPV-2008]: Lattice trapdoor. History-free reduction in ROM \implies QRROM.
- [BL-2016]: Lattice trapdoor. Standard model (no ROM).
- Trapdoor sigs tend to be inefficient in practice.
- [AFLT-2012]: A variant of the Lyubashevsky scheme.
 - ROM but not QRROM (due to AFLT).
 - Still somewhat inefficient.

Comparison: LWE/SIS schemes

- The only other scheme with parameters selected according to a reduction from LWE/SIS is the trapdoor scheme [GPV-2008]. (Parameters and implementation in [BB-2013].)
- Compared to [GPV-2008] at 96-bit classical, 59-bit quantum security:
 - TESLA-2 key sizes 25% smaller.
 - TESLA-2 sig sizes 87% smaller.
 - TESLA-1 cycle counts < 50% smaller.
- Ring-[GVP-2008] is more efficient, but so too would be Ring-TESLA.

Comparison: hash-based schemes

- The only other QR schemes with security reductions in the QRROM are hash-based schemes. (*e.g.* SPHINCS, Leighton-Micali.)
 - TESLA key sizes are much larger.
 - TESLA cycle counts are larger, but could become smaller with future work. (Ring-TESLA.)
 - TESLA signature size is much smaller.